计 算 机 研 究 与 发 展
Journal of Computer Research and Development

https://crad.ict.ac.cn
63(6): 1522−1533, 2026

基于大语言模型的运动模式感知轨迹表示学习

周泽宇[1,2] 林 彦[3] 郭晟楠[1,2] 林友芳[1,2] 万怀宇[1,2]

[1](北京交通大学计算机科学与技术学院 北京 100044)

[2](交通数据挖掘与具身智能北京市重点实验室(北京交通大学) 北京 100044)

[3](奥尔堡大学计算机科学系 丹麦奥尔堡 9220)

(zeyuzhou@bjtu.edu.cn)

Leveraging LLM for Movement Patterns-Aware Trajectory Representation Learning

Zhou Zeyu[1,2], Lin Yan[3], Guo Shengnan[1,2], Lin Youfang[1,2], and Wan Huaiyu[1,2]

[1] (*School of Computer Science and Technology*, *Beijing Jiaotong University*, *Beijing* 100044)

[2] (*Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence* (*Beijing Jiaotong University*), *Beijing* 100044)

[3] (*Department of Computer Science*, *Aalborg University*, *Aalborg*, *Denmark* 9220)

Abstract Spatial-temporal trajectory mining plays a crucial role in location-based services. Trajectory representation learning transforms complex trajectory features into low-dimensional representation vectors, enabling various downstream tasks. This process requires effectively extracting spatial-temporal features and motion patterns from trajectories. However, trajectory representation learning faces numerous challenges due to limitations in model capacity and issues with the scale of trajectory datasets. Large language model (LLM), pre-trained on massive high-quality datasets, have demonstrated remarkable capabilities in multi-task applications. Given the similarities between trajectories and natural language, it is feasible to develop an efficient trajectory representation method leveraging LLM. Nevertheless, vanilla LLMs cannot directly handle the unique spatial-temporal characteristics of trajectories or effectively extract motion patterns. To address these issues, we propose a trajectory representation learning model, MPTraj, which leverages LLMs to extract motion patterns. MPTraj introduces a novel trajectory semantic embedder that enables LLMs to process spatial-temporal features and extract movement patterns effectively. Additionally, a trajectory encoder with prompt tuning is integrated to generate high-quality trajectory representations. Furthermore, two pre-training tasks—next trajectory point prediction and movement pattern classification—are designed to enhance the model's performance on downstream tasks. Experimental results on two real-world datasets and four representative tasks demonstrate the effectiveness of MPTraj.

Key words spatial-temporal data mining; trajectory representation learning; trajectory embedding; large language model; pretrain-finetune

摘 要 时空轨迹挖掘在基于位置的服务中具有重要作用。轨迹表示学习将复杂的轨迹特征表示为低维表示向量,从而可以服务于多种下游任务,这一过程需要有效地从轨迹中提取时空特征和运动模式。然而,由于模型容量的局限以及轨迹数据集的规模问题,轨迹表示学习面临诸多挑战。大语言模型(large language model, LLM)通过在大规模、高质量数据集上的预训练,展现了其在多任务应用上的强大能力。鉴于轨迹与自然语言之间存在一定的相似性,利用 LLM 开发一种高效的轨迹表示方法成为可能。然而

收稿日期: 2025-05-26; **修回日期**: 2025-12-08

基金项目: 北京市自然科学基金项目(4242029)

This work was supported by Beijing Natural Science Foundation (4242029).

通信作者: 万怀宇(hywan@bjtu.edu.cn)

原始 LLM 并不能直接处理轨迹的独特时空特性，也无法有效提取运动模式。为了解决这些问题，提出了一种利用 LLM 提取运动模式的轨迹表示学习模型 MPTraj（movement patterns-aware trajectory representation via LLM），该模型引入了一种新颖的轨迹语义嵌入器，使得 LLM 能够处理时空特性并提取运动模式，同时结合基于提示微调的轨迹编码器生成高质量的轨迹表示。此外，设计了 2 个预训练任务，即下一轨迹点预测和运动模式分类，来提升模型在下游任务中的表现。在 2 个真实世界数据集和 4 个代表性任务上的实验结果验证了 MPTraj 的有效性。

关键词 时空数据挖掘；轨迹表示学习；轨迹嵌入；大语言模型；预训练微调

中图法分类号 TP391

DOI: 10.7544/issn1000-1239.202550354 **CSTR:** 32373.14.issn1000-1239.202550354

时空轨迹（spatial-temporal trajectory）是由一系列带有时间戳的位置组成的序列，表示为$\mathcal{T}=\{(l_1,t_1),(l_2,t_2),\cdots,(l_n,t_n)\}$，记录个体或物体在地理空间中的移动过程。随着移动电话、汽车导航系统、基于位置的服务和在线地图服务的广泛应用，时空轨迹正从各种来源中被记录和收集[1]。这些轨迹为一系列时空数据挖掘任务和应用提供了支持，包括轨迹预测[2-3]、异常检测[4-5]、轨迹相似性度量[6-9]以及轨迹用户关联[10-11]。

为了在各种任务和应用中更好地利用时空轨迹，近年来轨迹表示学习（trajectory representation learning）应运而生[8,12]。轨迹表示学习，即使用轨迹编码器将轨迹表示为d维向量，并将该表示应用于各种下游任务。为获取良好的性能，轨迹编码器需要：1）有效捕捉轨迹中蕴含的信息，特别是描述个体或物体如何从一个位置移动到另一个位置的运动模式；2）准确执行多种下游任务，减少为每个任务和应用设计单独方法的需求。现有的研究大多遵循自监督表示学习方法[13-14]，通过将轨迹映射为嵌入向量从零开始训练轨迹编码器[12,15-16]。然而，轨迹数据固有的高维时空特性与跨任务通用模型的构建难度共同制约了模型性能的进一步提升。这一局限性主要源于 2 方面因素：模型架构的容量限制，以及现有轨迹数据集在数据规模、标注质量和场景覆盖度方面的不足。

另一方面，通用预训练模型在自然语言处理（natural language processing, NLP）领域展现出强大的多任务适应能力，近年来大语言模型（large language model, LLM）在各种下游任务上都展现了可喜的成果[14,17-19]，这些模型通过融合大规模语料训练、深层架构设计以及提示工程优化，成功实现了语义理解与生成能力的跃升[20]。值得注意的是，时空轨迹与自然语言在结构特征上存在显著相似性：轨迹点间的时空关联性可类比词语的上下文依赖关系；运动模式，如转向角度、加速度变化等蕴含的物理语义与词汇的语法功能相呼应。这种结构相似性为迁移 LLM 的强大表征能力构建轨迹学习模型提供了理论依据，但在实际应用中仍面临两大核心挑战：

1）LLM 无法直接处理轨迹中的时空特征。LLM 本质上以离散的词语序列作为输入处理单元，而轨迹数据本质上是时空连续性与离散性并存的复杂系统。具体而言，轨迹特征包含连续型变量，如 GPS 坐标的经纬度数值和离散型变量，如时间戳、道路段编码的混合表征。如何将这些时空特征转化为适合 LLM 处理的序列，同时保留其物理意义和时空关联性，是当前面临的重要挑战。

2）LLM 无法直接从轨迹中提取运动模式。轨迹中的运动模式通过轨迹点特征的变化体现，图 1 展示了一段真实的车辆轨迹，在①段车辆航向角变化不大，速度不快且持续降低，那么可以认为该段有“直行”“减速”和“慢”的运动模式，同理车辆处于②段时以较慢的速度匀速转弯，在③段时速度较快并继续加速直行。而这些运动模式可以通过车辆位置和



Fig. 1 A spatial-temporal trajectory and its associated movement patterns

图 1 一段时空轨迹及其相关的运动模式

时间戳等特征的变化值计算并量化。但 LLM 并不具备直接处理连续特征的能力，也无法从轨迹中提取出运动模式。

为了解决这些挑战，并有效利用 LLM 构建轨迹表示学习模型，本文提出了一种名为 MPTraj(movement patterns-aware trajectory representation via LLM)的新方法。MPTraj 包含一个轨迹语义嵌入器(trajectory semantic embedder)，使 LLM 能处理轨迹中的时空特征，并有效提取运动模式。MPTraj 通过引入软提示提取轨迹的运动模式并生成轨迹表示。为了增强 MPTraj 的性能，本文还设计了下一个轨迹点预测和运动模式分类的预训练任务。

本文的主要贡献可总结如下：

1) 提出了 MPTraj 模型，该模型有效迁移 LLM 来处理轨迹时空特征和提取运动模式。通过提示微调，MPTraj 能生成准确且通用的轨迹表示，并应用于各种下游任务中，缓解了轨迹数据集规模小的限制。

2) 提出了一种新颖的轨迹语义嵌入器，使得 LLM 能够处理轨迹中的时空特征，并以可解释的方式有效提取了轨迹运动模式。

3) 在 2 个真实轨迹数据集上进行了广泛实验，通过各种实验设置验证了所提出模型的有效性。结果表明，MPTraj 作为一个轨迹表示学习模型，在不同任务中表现出了强大的性能。

1 相关工作

1.1 轨迹表示学习

轨迹表示的质量对于实现良好的轨迹分析任务性能至关重要。在很多专门针对单一任务进行端到端训练的任务中，轨迹被映射为轨迹表示，随后被输入预测网络完成下游任务[2,6,21]。然而，这样生成的轨迹表示通常不具备泛化性，难以迁移到其他模型或任务。

为了解决这一问题，越来越多的工作使用自监督轨迹表示学习。例如，t2vec[8] 构建了一个基于 RNN 的轨迹编码器，并将其应用于 GPS 序列，使用去噪自编码器学习轨迹表示。GM-VSAE[5] 进一步将轨迹映射到高斯潜在空间，以便对轨迹的分布进行建模。Traj2vec[22] 则考虑了相对的时空属性，使用连续轨迹点之间的一组属性差异作为输入特征。TremBR[12] 还将轨迹映射在道路网络上来建模出行语义，并将每个点的行程时间与嵌入向量拼接，结合了轨迹点的时间信息。CTLE[23] 采用双向 Transformer，并使用 2 个掩码重建(MLM)任务进行位置和时间预测，Toast[24] 利用 node2vec[25] 生成路段嵌入，并在 MLM 任务的基础上增加了序列判别任务。此外还有一些基于对比学习的方法，如 TrajCL[26] 提出了几种轨迹增强方法，并应用 MoCo 进行模型训练；MMTEC[27] 使用最大熵编码[28] 对离散和连续属性进行对比提取；START[29] 结合了 MLM 和 SimCLR 两种自监督任务来增强其学习能力；LightPath[15] 则结合了重构任务和隐藏空间上的对比任务。

这些方法由于是自监督并从头开始训练，其性能依赖于训练数据集的规模和质量，而这往往存在局限性。尽管现有方法取得了一定的成就，但仍需进一步努力以提升轨迹表示模型的性能。

1.2 LLM 的跨域应用

LLM 在自然语言处理领域的多功能性和卓越性能促使研究者尝试在其他领域中采用 LLM 以提高性能。在时间序列分析中，GPT4TS[30] 先对时间序列按片划分并进行块归一化，然后通过冻结自注意力前馈层来使用 LLM 处理块序列。Time-LLM[31] 引入了一种重编码框架，将时间序列表示成多个词包的线性组合，并使用全冻结的 Transformer 编码层来处理表示后的序列。LM4VisualEncoding[32] 将 LLM 的冻结 Transformer 块作为通用视觉编码层进行整合。RLMRec[33] 通过对齐框架，将 LLM 的语义空间与协作关系信号相结合。

尽管这些研究提供了宝贵的见解，但其方法不能直接应用于轨迹学习。轨迹数据具有独特的时空特征，需要特定的方法和考虑。

2 预备知识

本节将对本文所涉及的基本概念以及要解决的问题进行形式化定义。

2.1 基本定义

定义 1. 路网(road network)。道路网络通常表示为一个有向图$\mathcal{G}=(\mathcal{V},\mathcal{E})$。其中$\mathcal{V}$是包含$|\mathcal{V}|$个顶点的集合，每个顶点$v_i\in\mathcal{V}$表示道路段之间的交叉口或某一段道路的端点；$\mathcal{E}$是包含$|\mathcal{E}|$条边的集合，每条边$s_i\in\mathcal{E}$表示连接 2 个顶点的道路段。

定义 2. 时空轨迹(spatial-temporal trajectory)。时空轨迹$\mathcal{T}$是一系列带有时间戳的位置点的序列，表示为$\mathcal{T}=\{(l_1,t_1),(l_2,t_2),\cdots,(l_n,t_n)\}$。其中每个位置$l_i$由其经纬度坐标表示，即$l_i=\left(l_i^{\text{lat}},l_i^{\text{lng}}\right)$。时间戳$t_i$表示访问位置$l_i$的时间。为简化表示，我们将第$i$个轨迹点

(l_i, t_i)记为τ_i。

2.2 问题定义

轨迹表示学习的目标是开发1个轨迹编码器f_Θ，其中包含1组可学习的参数Θ。该模型将轨迹$\mathcal{T}$作为输入，输出轨迹的低维表示$\boldsymbol{z} \in \mathbb{R}^d$，即$\boldsymbol{z} = f_\Theta(\mathcal{T})$。本研究关注的重点是基于LLM训练一个轨迹编码器，目标是获得在各种下游任务中具有适应性的嵌入。具体来说，所学习的嵌入旨在提高不同轨迹挖掘任务中的收敛速度和准确性，使其在众多应用中成为有效的工具。

2.3 大语言模型

在本研究中，LLM指的是基于Transformer的语言模型，该模型在语料数据集上经过预训练。LLM包括4个核心模块，形式化地表示为

$$\mathrm{LLM} = \mathrm{LMHead} \circ \mathrm{TransBlk} \circ \mathrm{WTE} \circ \mathrm{Tok}, \quad (1)$$

其中符号$\circ$表示函数的复合操作。具体而言，LLM包括以下模块：1)分词器(tokenizer, Tok)，用于将文本分解为离散的tokens；2)嵌入层WTE(work token embedding)，将tokens转换为数值向量以捕捉其语言学特征；3)Transformer层TransBlk(Transformer block)，进一步处理这些向量以捕捉它们的上下文关系；4)预测头LMHead(language model head)，用于执行具体的预测任务，例如生成序列中的下一个单词。

3 运动模式感知的轨迹表示学习模型

本节将具体介绍基于LLM的运动模式感知的轨迹表示学习方法MPTraj。图2展示了MPTraj的整体框架，其实现过程可以分为如下3个步骤：

1)轨迹语义嵌入器。对于给定的轨迹$\mathcal{T}$，首先执行地图匹配以扩展轨迹特征。随后通过时空嵌入模块和模式语义投影模块将轨迹映射为d维嵌入序列$\boldsymbol{Z}_\mathcal{T}$。轨迹语义嵌入器使得LLM能够处理时空特征，同时有效提取轨迹中的运动模式。

2)基于LLM的轨迹编码器。通过引入虚拟tokens，并利用提示编码器生成连续提示。这些提示与嵌入后的轨迹序列拼接后，输入到冻结参数的LLM模块TransBlk中进行处理，最终通过模型最后位置的输出向量生成轨迹表示向量。

3)预训练任务。在模型应用于下游任务之前，采用下一个轨迹点预测和运动模式分类2个预训练任务。这种预训练方式能够提升轨迹表示的准确性和模型的泛化能力。

接下来将对MPTraj模型进行详细的描述。

3.1 轨迹语义嵌入器

如图1所示，运动模式可以通过运动对象在道路

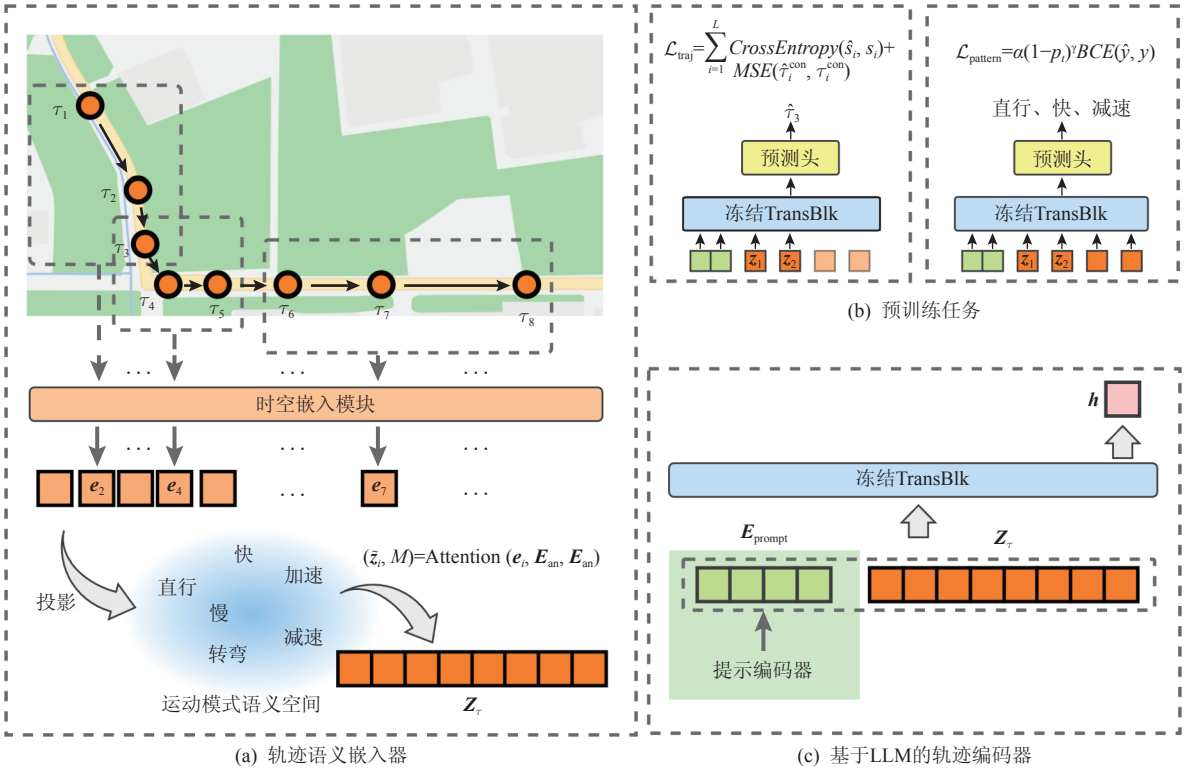


Fig. 2 The framework of MPTraj model

图2 MPTraj模型框架

网络上的位置及其时空特征的变化进行描述。为了使 LLM 能够处理轨迹中的时空特征并捕获运动模式，本文提出了一种轨迹语义嵌入器，其结构如图 2(a)所示。

轨迹语义嵌入器首先进行道路匹配，对轨迹特征进行扩展。接着，通过设计的时空嵌入模块对轨迹点进行嵌入，生成每个轨迹点的时空特征表示。最后，利用注意力机制将这些时空特征投影到运动模式语义空间中，从而实现对运动模式的高效提取。

3.1.1 轨迹特征扩展

为了提取轨迹点在道路网络上的路段信息，MPTraj 先使用 LMM (leuven map matching)道路匹配算法将每个轨迹点τ_i映射到道路网络$\mathcal{G}$上，得到：

$$\mathrm{LMM}(\mathcal{T},\mathcal{G})=\{(l_1,s_1,t_1),(l_2,s_2,t_2),\cdots,(l_n,s_n,t_n)\}, \tag{2}$$

其中s_i表示l_i所在的道路段，$i\in\{1,2,\cdots,n\}$。

3.1.2 时空嵌入模块

模块将扩展后轨迹的时空特征嵌入到一个d维特征空间中。对于离散的道路段s_i，我们采用嵌入表查找的嵌入方式$\boldsymbol{E}_{\mathcal{E}}\in\mathbb{R}^{|\mathcal{E}|\times d}$，其中道路$s_i$的嵌入向量表示为$\boldsymbol{E}_{\mathcal{E}}(s_i)$。对于时间戳$t_i$，我们利用 2 个嵌入表$\boldsymbol{E}_{\mathrm{d}}\in\mathbb{R}^{7\times d}$和$\boldsymbol{E}_{\mathrm{h}}\in\mathbb{R}^{24\times d}$，分别嵌入表示 1 周中的天(day-in-week)和小时(hour)的循环时间特征，嵌入表示为$\boldsymbol{E}_{\mathrm{d}}(t_i)$和$\boldsymbol{E}_{\mathrm{h}}(t_i)$。

为了从连续特征的变化中建模运动模式，时空嵌入模块采用 1 维卷积网络对连续特征进行嵌入。对于第i个轨迹点的连续特征$\tau_i^{\mathrm{con}}=\left(l_i^{\mathrm{lat}},l_i^{\mathrm{lng}},t_i\right)$，卷积操作定义为：

$$\boldsymbol{E}_{\mathrm{con}}(i)=Conv1D\left(\tau^{\mathrm{con}}_{i-\lfloor\frac{K}{2}\rfloor:\,i+\lfloor\frac{K}{2}\rfloor}\right), \tag{3}$$

其中K是超参数，表示卷积核的大小，$\boldsymbol{E}_{\mathrm{con}}(i)\in\mathbb{R}^d$表示轨迹点$\tau_i$的连续特征嵌入向量。

最后，第i个轨迹点τ_i的嵌入向量表示为

$$\boldsymbol{e}_i=\boldsymbol{E}_{\mathrm{con}}(i)+\boldsymbol{E}_{\mathcal{E}}(s_i)+\boldsymbol{E}_{\mathrm{d}}(t_i)+\boldsymbol{E}_{\mathrm{h}}(t_i)\ 。 \tag{4}$$

3.1.3 运动模式语义投影器

为了增强 LLM 对轨迹运动模式的理解能力，本节设计了一种模式语义投影器(pattern semantic projector, PSP)，该投影器将每个嵌入向量$\boldsymbol{e}_i$投影到一个语义丰富的文本空间中。

文本空间由一组用于描述运动模式的词汇定义，这些词汇包括直行(straight)、转弯(turn)、加速(accelerate)、减速(deaccelerate)、匀速(stable)、快(fast)和慢(slow)，它们被称为锚点词(anchor words)。通过式(1)中引入的 LLM 组件WTE$\circ$Tok获取锚点词的嵌入向量，形式化地表示为$\boldsymbol{E}_{\mathrm{an}}\in\mathbb{R}^{7\times d}$。

为了将嵌入向量$\boldsymbol{e}_i$投影到由$\boldsymbol{E}_{\mathrm{an}}$张成的空间，本节采用了点积注意力机制。注意力的计算使用$\boldsymbol{e}_i$作为查询，$\boldsymbol{E}_{\mathrm{an}}$作为键和值，其形式化地表示为

$$(\tilde{\boldsymbol{z}}_i,\boldsymbol{M})=Attention(\boldsymbol{e}_i,\boldsymbol{E}_{\mathrm{an}},\boldsymbol{E}_{\mathrm{an}}), \tag{5}$$

其中$\tilde{\boldsymbol{z}}_i$为注意力机制的输出向量，$\boldsymbol{M}$是注意力计算中的分数矩阵，具体计算公式为：

$$\boldsymbol{M}=softmax\left(\frac{\boldsymbol{Q}(\boldsymbol{E}_{\mathrm{an}})^{\mathrm{T}}}{\sqrt{d}}\right), \tag{6}$$

其中$\boldsymbol{Q}$为所有$\boldsymbol{e}_i$组成的矩阵。

最后通过残差连接获得每个轨迹点最终的嵌入向量$\boldsymbol{z}_i$：

$$\boldsymbol{z}_i=MLP(\tilde{\boldsymbol{z}}_i)+\boldsymbol{e}_i, \tag{7}$$

其中MLP表示一个 2 层的全连接网络。

最终，轨迹$\mathcal{T}$的嵌入序列为$\boldsymbol{Z}_{\mathcal{T}}=(\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_L)$，其中$L$是轨迹序列的长度。

3.2 基于 LLM 的轨迹编码器

为了保留 LLM 通用的特征建模能力，将 LLM 中的 Transformer 模块TransBlk作为所提出的基于 LLM 的轨迹编码器(trajectory encoder with LLM, TEL)的主干网络，并冻结其所有参数。为了更好地将TransBlk适配于时空轨迹数据，TEL 引入了提示微调(prompt tuning)。具体而言，在轨迹嵌入序列输入模型前，TEL 将其与一段可学习的连续提示拼接。

TEL 首先引入N_v个虚拟 tokens，并随机初始化这些虚拟 tokens 的嵌入向量$\tilde{\boldsymbol{E}}_{\mathrm{prompt}}\in\mathbb{R}^{N_v\times d}$，其中$N_v$是超参数。随后，通过一个 2 层全连接网络对这些虚拟 tokens 进行进一步处理，生成新的嵌入向量$\boldsymbol{E}_{\mathrm{prompt}}\in\mathbb{R}^{N_v\times d}$：

$$\boldsymbol{E}_{\mathrm{prompt}}=MLP\left(\tilde{\boldsymbol{E}}_{\mathrm{prompt}}\right)\ 。 \tag{8}$$

接着，将$\boldsymbol{E}_{\mathrm{prompt}}$与轨迹嵌入序列$\boldsymbol{Z}_{\mathcal{T}}$拼接后，输入到冻结参数的TransBlk中。随后，仅输出轨迹序列$\boldsymbol{Z}_{\mathcal{T}}$对应位置的输出向量$\boldsymbol{H}\in\mathbb{R}^{L\times d}$，形式化地表示为：

$$\boldsymbol{H}=TEL(\boldsymbol{Z}_{\mathcal{T}})=TransBlk(\boldsymbol{E}_{\mathrm{prompt}}\|\boldsymbol{Z}_{\mathcal{T}})[|N_v|:\,]\ 。 \tag{9}$$

最后，取$\boldsymbol{H}$的最后一位向量作为最终的轨迹表示向量$\boldsymbol{h}$。

3.3 模型预训练

为了提高轨迹表示的准确性和泛化性，本节提出了 2 个模型预训练任务：下一个轨迹点预测任务和运动模式分类任务。

3.3.1 下一个轨迹点预测任务

下一个轨迹点预测任务采用自回归的方式重建轨迹特征，如图 2(c)所示。对于给定的轨迹$\mathcal{T}$，重构过程包括$|\mathcal{T}|$步。在第i步中，TEL 接收由$\boldsymbol{E}_{\mathrm{prompt}}$和前

$i-1$个轨迹点嵌入：

$$\boldsymbol{H}_{\text{traj},i-1}=TEL(\boldsymbol{E}_{\text{prompt}}\|\boldsymbol{Z}_{\mathcal{T},:i-1})。\tag{10}$$

然后，通过将$\boldsymbol{H}_{\text{traj},i-1}$连接到预测头上来预测第$i$个轨迹点$\tau_i$的特征，预测头由2层全连接网络实现。轨迹点重构损失$\mathcal{L}_{\text{traj}}$由所在路段重构的交叉熵损失和预测的连续特征重构的均方差误差损失构成：

$$\mathcal{L}_{\text{traj}}=\sum_{i=1}^{L}CrossEntropy(\hat{s}_i,s_i)+MSE\left(\hat{\tau}_i^{\text{con}},\tau_i^{\text{con}}\right)，\tag{11}$$

其中$\hat{s}_i$为预测的所在路段特征，$\hat{\tau}_i^{\text{con}}$为预测的连续特征。

3.3.2 运动模式分类任务

运动模式分类任务与下一个轨迹预测任务同时进行，对模式语义投影器中的注意力分数矩阵$\boldsymbol{M}$进行监督。具体而言，运动模式分类任务是一个多标签分类任务，其中$\boldsymbol{M}_i\in\mathbb{R}^7$表示第$i$个轨迹点与锚点词的关联程度，该分数需要能正确预测以τ_i为中心、K为长度的轨迹片段的正确运动模式。

本任务首先分段处理轨迹标签，以时空嵌入模块卷积核长度K为滑动窗口，若窗口内航向角变化超过30°则标签为“转弯”，否则为“直行”；若窗口内平均速度超过50 km/h则标签为“快”，否则为“慢”；若窗口内加速度超过2.77 m/s^{2}则标签为“加速”，低于−2.77 m/s^{2}则标签为“减速”，否则为“匀速”。在损失函数设计上，该任务先通过Sigmoid函数将注意力分数映射到(0,1)之间，然后设置7组二分类实验。为了缓解样本不均衡问题，引入了焦点损失（focal loss）的设计，运动模式分类任务的损失$\mathcal{L}_{\text{pattern}}$可形式表达为：

$$\mathcal{L}_{\text{pattern}}=\alpha(1-p_t)^{\gamma}BCE(\hat{y},y)，\tag{12}$$

其中$BCE(\hat{y},y)$是预测值$\hat{y}$与真实标签y之间的二元交叉熵损失，$p_t=\mathrm{e}^{-BCE(\hat{y},y)}$表示模型对正确类别的预测概率，$\alpha$是平衡因子，用于调整正负样本的权重并取值0.25，γ是聚焦参数，用于调整简单样本与困难样本的权重并取值2.0。

最终，预训练任务的总损失函数表示为

$$\mathcal{L}_{\text{pre}}=\mathcal{L}_{\text{traj}}+\mathcal{L}_{\text{pattern}}。\tag{13}$$

4 实　　验

在2个真实世界数据集4个任务上与7个基线模型进行实验对比，验证MPTraj模型的有效性。然后，设计消融实验来验证模型各组件的有效性，并通过分析超参数，深入探讨模型性能与关键参数设置之间的关系。

4.1 实验数据集

使用了2个真实的出租车轨迹数据集，分别是来自葡萄牙波尔图市和中国成都市的轨迹数据。其中波尔图（Porto）数据集记录了2013年7月到2014年6月期间波尔图全市出租车轨迹数据；成都（Chengdu）数据集记录了2018年10月1日至15日期间成都市所有滴滴出行网约车的轨迹数据。数据集统计信息概览如表1所示。

Table 1 Dataset Statistics

表1 数据集统计信息

项目	波尔图数据集	成都数据集
时间跨度	2013-07-01—2014-06-30	2018-09-30—2018-10-10
轨迹数目	538 495	140 000
路段数	13 695	4 315
记录数	18 832 411	22 330 589

4.2 轨迹数据挖掘下游任务

轨迹表示的有效性将在如下4个轨迹数据挖掘下游任务中进行验证分析：

1）轨迹时间估计（trajectory time estimation, TTE）。TTE任务旨在根据轨迹的空间特征和出发时间估计总旅行时间，而不使用包括时间戳、速度和加速度在内的时间相关特征。为此，我们使用一个2层全连接网络构建预测头，其预测公式为

$$\hat{y}_{\text{TTE}}=MLP_{\text{TTE}}(\boldsymbol{h})。\tag{14}$$

2）目的地预测（destination prediction, DP）。DP任务旨在预测轨迹的目的地所在的道路段，输入轨迹中排除最后5个点以避免数据泄露。为此，我们使用一个2层全连接网络构建预测头，其输出维度对应于道路段总数$|\mathcal{E}|$，预测公式为

$$\hat{y}_{\text{DP}}=\arg\max_{s}(\hat{p}),\hat{p}=softmax(MLP(\boldsymbol{h}))。\tag{15}$$

3）轨迹分类（trajectory classification, TC）。TC任务旨在将轨迹分配到正确的类别中。在本工作中，我们将与轨迹相关联的驾驶员作为分类任务的标签。为此，我们使用一个2层全连接网络构建预测头，其输出维度对应于驾驶员总数$|\mathcal{D}|$，预测公式为

$$\hat{y}_{\text{TC}}=\arg\max_{d}(\hat{p}),\hat{p}=softmax(MLP(\boldsymbol{h}))。\tag{16}$$

4）相似轨迹搜索（STS）。STS任务旨在从候选集中识别与给定查询最相似的轨迹。我们通过计算轨迹表示$\boldsymbol{h}$之间的余弦相似度来评估轨迹间的相似性，公式为

$$CosSim(\mathcal{T}_1,\mathcal{T}_2)=\frac{\boldsymbol{h}_1\cdot\boldsymbol{h}_2}{|\boldsymbol{h}_1||\boldsymbol{h}_2|}。\tag{17}$$

由于大多数数据集未提供此任务的真实标签，我们采用如下方法生成标签：首先从测试数据集中随机选取 1 000 条轨迹。对于每条轨迹$\mathcal{T}$，将其奇数索引点构成查询$\mathcal{T}^{\mathrm{q}}$，偶数索引点构成目标$\mathcal{T}^{\mathrm{t}}$。对于每个查询，排除距离查询最近的前 10 条轨迹，然后从剩余测试集中随机选择 5 000 条轨迹组成数据库。在计算查询与其他轨迹的距离时，我们参考文献 [7] 的方法，先将轨迹下采样至统一长度，然后计算其均方根误差。

4.3 评价指标

在旅行时间预测的回归任务中，采用以下指标来衡量模型性能：

均方根误差(root mean square error, *RMSE*)，其计算公式为

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i-y_i)^2}。\tag{18}$$

平均绝对误差(mean absolute error, *MAE*)，其计算公式为

$$MAE=\frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i-y_i|。\tag{19}$$

平均绝对百分比误差(mean absolute percentage error, *MAPE*)，其计算公式为

$$MAPE=\frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i-y_i}{y_i}\right|,\tag{20}$$

其中$\hat{y}_i$为预测值，y_i为真实值。这些指标的值越小，模型性能越佳。

对于轨迹分类、目的地预测和相似轨迹预测的分类任务，使用以下指标：

准确率(*ACC@1*)，其计算公式为

$$ACC@1=\frac{\text{符合 Top-1 预测的样本数}}{\text{总样本数}}\times100\%。\tag{21}$$

Top-5 准确率(*ACC@5*)，其计算公式为

$$ACC@5=\frac{\text{符合 Top-5 预测的样本数}}{\text{总样本数}}\times100\%。\tag{22}$$

召回率(*Recall*)，其计算公式为

$$Recall=\frac{\text{正确预测为目标类别的样本数}}{\text{目标类别的总样本数}}\times100\%。\tag{23}$$

这些指标值越大，模型性能越优。此外，在相似轨迹搜索任务中，还引入平均排名(mean rank, *MR*)指标：

$$MR=\frac{1}{N}\sum_{i=1}^{N}Rank_i,\tag{24}$$

其中N是测试集中查询轨迹的总数，$Rank_i$表示第 i 个查询轨迹的正确相似轨迹在返回结果中的排名。

4.4 基线模型

选取了一些经典和最先进的轨迹表示学习方法与本文提出的方法进行对比，包括：

1)T2vec[8] 使用去噪自编码器通过从低采样轨迹重建原始轨迹来预训练轨迹编码器。

2)TremBR[12] 构建了一个基于长短期记忆网络的序列到序列(seq2seq)模型[34]，能够恢复输入轨迹的路段和时间。

3)CTLE[23] 使用双向 Transformer 进行预训练，采用位置和小时预测的 2 个掩码重建(MLM)任务。轨迹表示通过对点嵌入平均池化得到。

4)Toast[24] 使用具有上下文感知的 node2vec 模型生成路段表示，并设置 MLM 任务和序列判别任务训练轨迹编码器。

5)TrajCL[26] 引入了一种基于双特征的自注意力编码器，并通过 InfoNCE 损失以对比学习方式训练轨迹编码器。

6)START[29] 包括一个时间感知的轨迹编码器和一个考虑道路段之间转移的图注意力网络(GAT)。该模型通过一个 MLM 任务和一个基于 SimCLR 损失的对比任务进行训练。

7)LightPath[15] 构建一个稀疏路径编码器，并设置路径重建任务和跨视图与跨网络对比任务。

8)MMTEC[27] 结合了基于注意力的离散编码器和基于神经控制微分方程的连续编码器，通过基于对比学习方法的最大熵与训练任务进行训练。

4.5 实验设置

对于每个数据集，根据出发时间的先后顺序，将轨迹按 8∶1∶1 的比例划分为训练集、验证集和测试集。模型在训练集上进行训练，并在测试集上进行评估。预训练任务和基线模型均进行 20 轮的预训练，而下游任务预测头则依据验证集的表现进行早停。

选用 GPT2 作为基座大模型来开发模型，MPTraj 的最佳参数为$d=128, d_{\text{output}}=128, K=5, N_v=10$。这些参数是基于成都验证集上的目的地预测任务的 *ACC@1* 和 *Recall* 进行选择的。在模型训练过程中，采用 Adam 优化器，预训练任务的初始学习率设定为 1E−4，微调学习率设定为 0.001。

4.6 实验结果分析

表 2 和表 3 详细比较了 MPTraj 与基线模型在 4 个任务及 2 个数据集上的表现。由于波尔图数据集是等距采样的，可能造成数据泄露，因此在波尔图数据集上应特别注意。此外，由于成都数据集中每位司机的轨迹数量有限，轨迹分类任务仅在波尔图数据集上进行。结果显示，MPTraj 在各项任务中表现出色，证明了其作为一种先进的轨迹表示学习方法的有效性。

T2vec 和 TremBR 基于循环神经网络的自编码器或自回归框架，但未能充分捕捉关键的时空特征，导致其在下游任务中的表现不理想。CTLE 和 Toast 通过引入双向 Transformer 架构并结合 MLM 任务提升模型能力，但由于缺乏必要的连续特征，难以有效捕捉轨迹运动模式，从而影响整体性能。特别地，因为未能直接学习轨迹级别的表示，这些方法在相似轨迹搜索任务中的表现不佳。尽管 TrajCL, START, LightPath 通过对比学习策略改善了相似轨迹搜索任务的性能，但仍存在不足。TrajCL 缺乏学习细粒度信息的有效监督机制，而 START 和 LightPath 在提取运动模式时未能充分利用连续特征信息，使其在其他下游任务中的表现不尽如人意。MMTEC 同时建模了连续坐标和路段 2 个视角，但模型规模也限制了其效果。相比之下，MPTraj 利用 LLM 的强大表示能力，不仅有效缓解了轨迹数据集的固有限制，还能高效提取运动模式，这些优势使得 MPTraj 在多个下游任务中展现出卓越的性能。

4.7 消融实验

为了评估 MPTraj 中各组件的有效性，将完整模型与以下变体进行比较：

1）w/o Pattern。移除运动模式监督损失，仅进行下一个轨迹点预测的预训练任务。

Table 2 Overall Performance Comparison on Porto Dataset

表 2 在波尔图数据集上的总体性能比较

方法	轨迹分类			目的地预测			相似轨迹搜索		
	ACC@1/%	*ACC@5/%*	*Recall/%*	*ACC@1/%*	*ACC@5/%*	*Recall/%*	*MR*	*ACC@1/%*	*ACC@5/%*
T2vec	50.880	96.248	49.191	20.168	43.230	5.324	1.683	91.64	95.133
TremBR	55.300	95.581	50.321	20.472	43.974	5.626	2.051	89.908	93.504
CTLE	85.162	97.426	66.663	18.775	40.718	2.796	5.589	64.675	82.750
Toast	51.102	82.088	27.604	18.731	39.673	2.412	6.013	61.533	80.367
TrajCL	88.012	96.308	76.217	20.601	45.876	4.590	1.196	93.500	98.933
START	89.464	97.122	73.913	21.377	46.777	5.750	1.127	96.500	99.300
LightPath	88.294	93.272	86.241	22.092	47.131	5.728	5.678	83.067	90.578
MMTEC	89.476	97.425	86.835	22.078	48.295	6.055	1.238	92.445	96.180
MPTraj（本文）	**92.055**	**98.388**	**87.349**	**25.786**	**57.454**	**7.085**	**1.081**	**97.185**	**99.900**

注：黑体数值表示最优结果，下划线数值表示次优结果。

Table 3 Overall Performance Comparison on Chengdu Dataset

表 3 在成都数据集上的总体性能比较

方法	旅行时间估计			目的地预测			相似轨迹搜索		
	RMSE/s	*MAE*/s	*MAPE/%*	*ACC@1/%*	*ACC@5/%*	*Recall/%*	*MR*	*ACC@1/%*	*ACC@5/%*
T2vec	128.508	60.520	15.224	47.739	73.509	16.638	3.345	81.450	93.700
TremBR	125.535	57.965	13.964	48.987	72.082	17.010	4.659	83.980	89.880
CTLE	132.636	57.481	13.153	51.004	79.434	21.467	9.429	53.767	69.200
Toast	128.793	60.997	14.883	50.897	79.664	21.068	5.944	53.640	71.600
TrajCL	120.211	59.816	14.741	50.847	79.693	21.572	1.198	95.125	98.875
START	122.205	55.922	12.717	52.775	80.423	23.316	1.089	96.933	99.900
LightPath	119.230	55.614	12.760	49.154	78.587	20.660	27.266	74.267	86.100
MMTEC	118.776	54.502	12.094	52.853	79.551	25.684	1.225	94.703	97.190
MPTraj（本文）	**114.087**	**51.186**	**11.566**	**56.049**	**84.266**	**27.789**	**1.059**	**97.923**	**99.940**

注：黑体数值表示最优结果，下划线数值表示次优结果。

2)w/o Pretrain。移除所有预训练任务,在每个下游任务上进行端到端训练。

3)w/o Conv。移除轨迹语义嵌入器中的卷积操作,改用单层全连接层进行连续特征嵌入。

4)w/o PSP。移除模式语义投影器,直接使用$\boldsymbol{e}_i$替代轨迹点嵌入$\boldsymbol{z}_i$。

在成都数据集上测量了这些变体的性能,结果如表 4 所示。基于实验结果可以发现,对运动模式识别的监督和下一个轨迹点预测的预训练任务都对 MPTraj 的结果具有重要的促进作用。同时,替换卷积模块导致效果下降,表明卷积模块在融合时空连续特征和识别运动模式方面对最终结果起着重要作用。此外,去掉模式语义投影器后的性能低于完整模型,表明该投影器有助于 MPTraj 充分利用大型语言模型的能力,从而对最终性能产生积极影响。

Table 4 Performance Comparison of MPTraj Variants on Chengdu Dataset

表 4 MPTraj 变体在成都数据集上的性能比较

方法	旅行时间估计		目的地预测	
	MAE/s	*MAPE*/s	*ACC@1*/%	*Recall*/%
w/o Pattern	53.859	12.031	54.913	27.391
w/o Pretrain	54.064	12.214	54.128	26.298
w/o Conv	53.212	12.073	54.544	25.741
w/o PSP	54.589	12.343	54.956	26.935
MPTraj(本文)	**51.186**	**11.566**	**56.049**	**27.789**

注:黑体数值表示最优值。

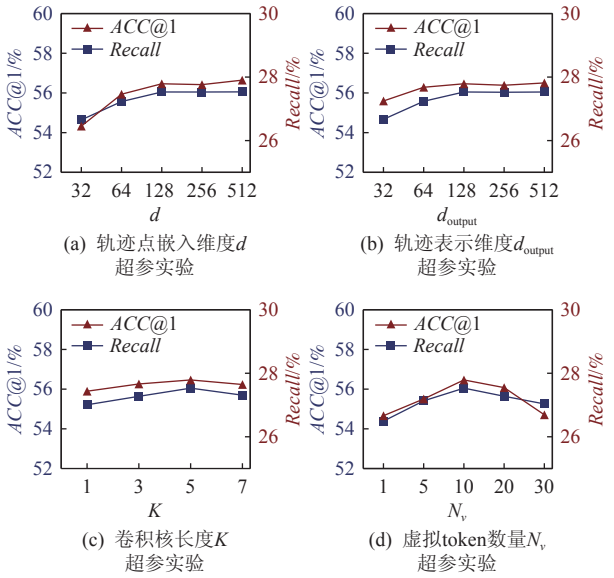
4.8 超参数分析

对 MPTraj 几个重要的超参数进行分析,包括轨迹语义嵌入器的嵌入维度d、卷积核长度K、语义投影模块虚拟 tokens 数量N_v,以及输出轨迹表示维度d_{output}。在目的地预测任务上评价超参数的影响并使用 *ACC@1* 和 *Recall* 这 2 个指标,结果如图 3 所示。

分析结果显示,随着嵌入维度d和轨迹表示维度d_{output}的增加,模型性能有所提升,但在超过 128 后,性能提升变得有限。因此,为了在性能和效率之间取得平衡,我们选择$d=128$和$d_{\text{output}}=128$作为最优参数。实验结果表明,卷积核长度K的最优值是 5,过小的卷积核无法识别到准确的运动模式,而过大的卷积核会引入噪声。同样地,虚拟 token 数量N_v的最优取值为 10,较小的长度限制了模型容量,而较大的N_v则导致过拟合问题。

4.9 案例分析

为了更加直观地展示模型如何准确且有效地提取运动模式,将模式语义投影器中的注意力分数可视化,如图 4 所示。



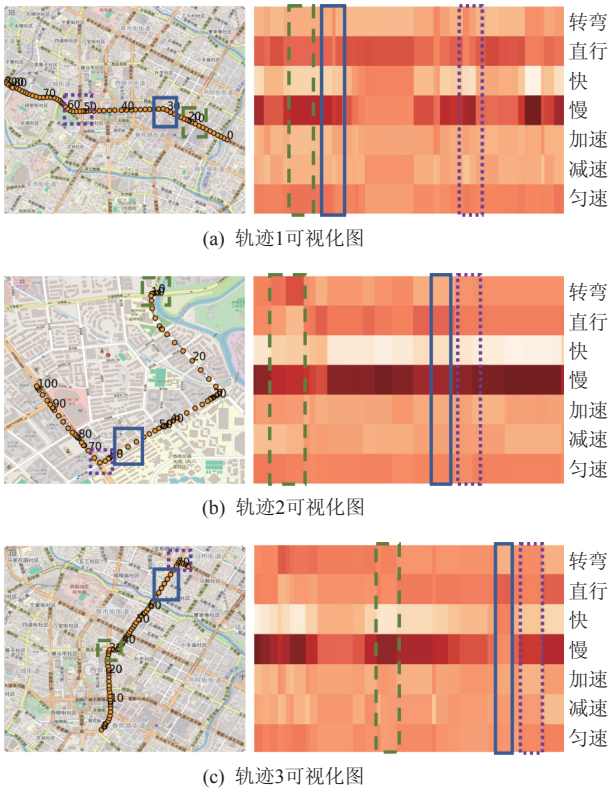
(a) 轨迹点嵌入维度d超参实验 (b) 轨迹表示维度d_{output}超参实验

(c) 卷积核长度K超参实验 (d) 虚拟token数量N_v超参实验

Fig. 3 Analysis diagram of hyper-parameters

图 3 超参数分析图

在每个示例中,原始轨迹显示在左侧,其中被 10 整除的索引点被标记出来。矩形框用于突出显示子轨迹。这些子轨迹的注意力图显示在右侧,用相同颜色来相互对应。



(a) 轨迹1可视化图

(b) 轨迹2可视化图

(c) 轨迹3可视化图

Fig. 4 Visualization of attention scores in PSP

图 4 模式语义投影器中的注意力分数可视化

如图 4(a)，短杠矩形框的轨迹片段直行，而实线和圆点矩形框的轨迹片段运动模式为转弯，这一变化被模式语义投影器所捕捉到。同样的，如图 4(b)，短杠矩形框和圆点矩形框的轨迹片段在加速，而实线矩形框的轨迹片段在减速，这一运动模式也反映在注意力热力图中。如图 4(c)，短杠矩形框的轨迹片段运动速度较慢，而实线矩形框的轨迹片段以较快速度直行，圆点矩形框的轨迹则在缓慢转弯，模式语义投影器也准确将这些运动模式捕获并投影至语义特征空间中。

4.10 TEL 解冻消融分析

为探究 TEL 解冻不同组件对模型性能的影响，本文将 MPTraj 与以下变体进行了对比分析：

1)ft LN。仅解冻 LLM Transformer 模块中的所有层归一化和位置编码层。

2)ft FFN。仅解冻 LLM Transformer 中的所有前向传播网络层。

3)ft all。对 LLM Transformer 的全部参数进行微调。

实验在成都数据集上的目的地预测任务中进行，结果如表 5 所示。可以发现，随着解冻组件数量的增加，模型在该任务上的性能并未提升。尤其是在解冻前向传播层和注意力层后，模型训练容易出现坍塌现象。综合考虑模型性能、稳定性及存储成本，仅微调提示编码器是最优选择。

Table 5 Performance Comparison of MPTraj Variants on Chengdu Dataset

表 5 MPTraj 变体在成都数据集上的性能比较

方法	目的地预测		可学习参数规模/MB
	ACC@1/%	*Recall*/%	
ft LN	55.34±1.05	26.78±2.27	22.43
ft FFN	36.62±29.97	16.77±14.47	245.36
ft all	20.40±29.50	8.49±14.46	500.78
MPTraj	**56.05±0.48**	**27.79±0.63**	19.28

注：黑体数值表示最优结果；下划线数值表示次优结果。

5 总结与展望

本文提出了一种创新的轨迹表示学习方法 MPTraj，该方法利用 LLM 来建模轨迹，并在多种下游任务中表现优异。MPTraj 引入了轨迹语义嵌入器，使 LLM 能够有效处理轨迹的时空特征，并提取目标的运动模式。该方法还通过提示微调技术增强了基于大模型的轨迹编码器。此外，为了提升模型的泛化性能和运动模式提取能力，MPTraj 提出了下一个轨迹点预测和运动模式分类 2 个预训练任务。在 2 个数据集上的 4 个下游任务及多种实验设置下的结果表明，MPTraj 具有显著的先进性。

未来，在本研究的基础上可以进一步探索更多数据驱动的运动模式建模方法，具体包括：扩展运动模式的语义空间，采用端到端的方式训练轨迹编码器；利用 LLM 自动识别运动模式，替代人工标注，获取更加丰富的运动模式类型；将 LLM 应用于相同运动模式的判别，结合对比学习等自监督方法，提升轨迹编码器对运动模式的表征能力。此外，还将进一步在更多不同场景的数据集上开展实验，以验证 MPTraj 框架的泛化能力，例如行人轨迹、动物迁徙轨迹等。

作者贡献声明：周泽宇提出了算法思路、实验方法，完成实验并撰写论文；林彦提出算法思路、实验方法并提出论文修改意见；郭晟楠提出指导意见并修改论文；林友芳提出指导意见并修改论文；万怀宇提出指导意见并修改论文。

参 考 文 献

[1] Zheng Yu, Wang Longhao, Zhang Ruochi, et al. GeoLife: Managing and understanding your past life over maps[C]//Proc of the 9th Int Conf on Mobile Data Management. Piscataway, NJ: IEEE, 2008: 211−212

[2] Feng Jie, Li Yong, Zhang Chao, et al. Deepmove: Predicting human mobility with attentional recurrent networks[C]//Proc of the 27th Int World Wide Web Conf. New York: ACM, 2018: 1459−1468

[3] Kong Dejiang, Wu Fei. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction[C]//Proc of the 27th Int Joint Conf on Artificial Intelligence. Morgan Kaufmann: IJCAI, 2018: 2341−2347

[4] Wang Haiquan, Feng Jiachen, Sun Leilei, et al. Abnormal trajectory detection based on geospatial consistent modeling[J]. IEEE Access, 2020, 8: 184633−184643

[5] Liu Yiding, Zhao Kaiqi, Cong Ga, et al. Online anomalous trajectory detection with deep generative sequence modeling[C]//Proc of the 36th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2020: 949−960

[6] Chao Cheng, Pu Feifan, Xu Jianqiu, et al. Efficient dimensionality reduction and query algorithm of trajectory data based on spatial position relation[J]. Journal of Computer Research and Development, 2024, 61(7): 1771−1790 (in Chinese)
(巢成, 蒲非凡, 许建秋, 等. 基于空间位置关系的轨迹数据高效降

维和查询算法[J]. 计算机研究与发展, 2024, 61(7): 1771−1790)

[7] Fang Ziquan, Du Yuntao, Zhu Xinjun, et al. Spatio-temporal trajectory similarity learning in road networks[C]//Proc of the 28th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2022: 347−356

[8] Li Xiucheng, Zhao Kaiqi, Cong Gao, et al. Deep representation learning for trajectory similarity computation[C]//Proc of the 34th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2018: 617−628

[9] Yang Peilun, Wang Hanchen, Zhang Ying, et al. T3s: Effective representation learning for trajectory similarity computation[C]//Proc of the 37th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2021: 2183−2188

[10] Zhou Fan, Gao Qiang, Trajcevski G, et al. Trajectory-user linking via variational AutoEncoder[C]//Proc of the 27th Int Joint Conf on Artificial Intelligence. Morgan Kaufmann: IJCAI, 2018: 3212−3218

[11] Miao Congcong, Wang Jilong, Yu Heng, et al. Trajectory-user linking with attentive recurrent network[C]//Proc of the 19th Int Conf on Autonomous Agents and Multiagent Systems. New York: ACM, 2020: 878−886

[12] 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): 1−25

[13] Dai A M, Le Q V. Semi-supervised sequence learning[C]//Proc of the 28th Advances in Neural Information Processing Systems. New York: ACM, 2015: 3079−3087

[14] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proc of the 2019 Conf of the North American Chapter of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2019: 4171−4186

[15] Yang Bin, Hu Jilin, Guo Chenjuan, et al. LightPath: Lightweight and scalable path representation learning[C]//Proc of the 29th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2023: 2999−3010

[16] Jiang Jiawei, Pan Dayan, Ren Houxing, et al. Self-supervised trajectory representation learning with temporal regularities and travel semantics[C]//Proc of the 39th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2023: 843−855

[17] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[EB/OL]. OpenAI Blog, 2019[2025-09-22]. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[18] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. Journal of Machine Learning Research, 2020, 21(140): 1−67

[19] Chen Xuanting, Ye Junjie, Zu Can, et al. Robustness of GPT large language models on natural language processing tasks[J]. Journal of Computer Research and Development, 2024, 61(5): 1128−1142 (in Chinese)
(陈炫婷, 叶俊杰, 祖璨, 等. GPT 系列大语言模型在自然语言处理任务中的鲁棒性[J]. 计算机研究与发展, 2024, 61(5): 1128−1142)

[20] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[C]//Proc of the 33rd Advances in Neural Information Processing Systems. New York: ACM, 2020: 1877−1901

[21] Wang Dong, Zhang Junbo, Cao Wei, et al. When will you arrive? Estimating travel time based on deep neural networks[C]//Proc of the 32nd AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2018: 2500−2507

[22] Yao Di, Zhang Chao, Zhu Zhihua, et al. Trajectory clustering via deep representation learning[C]//Proc of the 2017 Int Joint Conf on Neural Networks. Piscataway, NJ: IEEE, 2017: 3880−3887

[23] Lin Yan, Wan Huaiyu, Guo Shengnan, et al. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction[C]//Proc of the 35th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2021: 4241−4248

[24] Chen Yile, Li Xiucheng, Cong Gao, et al. Robust road network representation learning: When traffic patterns meet traveling semantics[C]//Proc of the 30th ACM Int Conf on Information & Knowledge Management. New York: ACM, 2021: 211−220

[25] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[C]//Proc of the 22nd ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2016: 855−864

[26] Chang Yanchuan, Qi Jianzhong, Liang Yuxuan, et al. Contrastive trajectory similarity learning with dual-feature attention[C]//Proc of the 39th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2023: 2933−2945

[27] Lin Yan, Wan Huaiyu, Guo Shengnan, et al. Pre-training general trajectory embeddings with maximum multi-view entropy coding[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 36(12): 9037−9050

[28] Liu Xin, Wang Zhongdao, Li Yali, et al. Self-supervised learning via maximum entropy coding[C]//Proc of the 35th Advances in Neural Information Processing Systems. New York: ACM, 2022: 34091−34105

[29] Jiang Jiawei, Pan Dayan, Ren Houxiang, et al. Self-supervised trajectory representation learning with temporal regularities and travel semantics[C]//Proc of the 39th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2023: 843−855

[30] Zhou Tian, Niu Peisong, Wang Xue, et al. One fits all: Power general time series analysis by pretrained LM[C]//Proc of the 36th Advances in Neural Information Processing Systems. New York: ACM, 2023: 43322−43355

[31] Jin Ming, Wang Shiyu, Ma Lintao, et al. Time-LLM: Time series forecasting by reprogramming large language models[J]. arXiv preprint, arXiv: 2310.01728, 2023

[32] Pang Ziqi, Xie Ziyang, Man Yunze, et al. Frozen transformers in language models are effective visual encoder layers[J]. arXiv preprint, arXiv: 2310.12973, 2023

[33] Ren Xubin, Wei Wei, Xia Lianghao, et al. Representation learning with large language models for recommendation[C]//Proc of the 33rd Int World Wide Web Conf. New York: ACM, 2024: 3464−3475

[34] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Proc of the 27th Advances in Neural Information Processing Systems. New York: ACM, 2014: 3104−3112



Zhou Zeyu, born in 2001. Master candidate. His main research interests include spatial-temporal data mining and deep learning.

周泽宇, 2001 年生。硕士研究生。主要研究方向为时空数据挖掘、深度学习。



Lin Yan, born in 1998. PhD. Member of CCF. His main research interests include spatial-temporal data mining and representation learning.

林　彦, 1998 年生。博士。CCF 会员。主要研究方向为时空数据挖掘、表示学习。



Guo Shengnan, born in 1992. PhD, associate professor. Member of CCF. Her main research interests include spatial-temporal data mining and deep learning.

郭晟楠, 1992 年生。博士，副教授。CCF 会员。主要研究方向为时空数据挖掘、深度学习。



Lin Youfang, born in 1971. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include data mining, machine learning, reinforcement learning, complex network, and intelligent technology and system.

林友芳, 1971 年生。博士，教授，博士生导师。CCF 高级会员。主要研究方向为数据挖掘、机器学习、强化学习、复杂网络、智能技术与系统。



Wan Huaiyu, born in 1981. PhD, professor, PhD supervisor. Distinguished member of CCF. His main research interests include spatial-temporal data mining, information extraction, and social networks mining.

万怀宇, 1981 年生。博士，教授，博士生导师。CCF 杰出会员。主要研究方向为时空数据挖掘、信息抽取、社交网络挖掘。