



# **More Than Routing: Joint GPS and Route Modeling for Refine Trajectory Representation Learning**



2025/4/14

# I 介绍

## 轨迹表示学习的重要性

轨迹表示学习在支持多种下游任务方面发挥着关键作用。这些任务包括旅行时间估计、轨迹分类和Top-k相似轨迹查询等。

轨迹数据中蕴含着丰富的时空信息，这些信息可以应用于城市规划、城市应急管理、传染病防控和智能物流系统等领域。

### 模型挑战

- GPS轨迹的不确定性**：GPS轨迹中存在大量的冗余和噪声信号，它们会严重影响模型的计算效率和性能。
- 路径轨迹的时空相关性**：路径具有复杂的时空相关性，物体经历路段换位时必须考虑路网的拓扑结构，而路段的行程时间与历史交通模式和当前行程状态有关。
- 信息融合的复杂性**：虽然GPS轨迹和路径轨迹描述的是同一个概念，由于两个数据源的视角不同，这意味着两个领域，融合不同领域的信息是一个挑战。

### 主要贡献

- 提出了一种基于多模态融合思想的轨迹表示学习框架JGRM**：该框架由层次化的GPS编码器、路径编码器和模态交互器组成，其中层次化的GPS编码器用于描述道路实体的特征，路径编码器用于考虑轨迹的时空相关性，模态交互器用于信息融合。
- 设计了两个可推广到后续研究工作的自监督任务用于模型训练**：MLM用于重构轨迹本身的时空连续性，CMM用于融合不同视角的机动性信息。
- 在两个真实数据集成都、西安上的大量实验验证了JGRM在各种设置下都能获得最佳性能。**

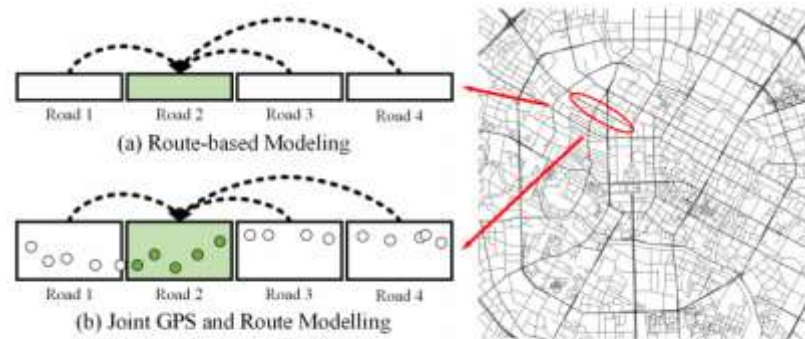





Figure 1: Route Modeling v.s. Fusion Modeling.


路线轨迹中的路段只能通过前面和后面的路段进行建模；GPS轨迹中的路段提供了更丰富的采样信息，允许对路段实体进行细粒度表示


# I 介绍

 **早期研究方法：**基于为特定下游任务设计的序列模型，并使用特定任务损失进行训练。这些表示缺乏泛化能力，难以应用于其他任务

 **seq2seq方法：**为解决表示的泛化问题，通过重构损失进行训练，以生成更通用的表示

 **图神经网络方法：**随着图神经网络的兴起，研究人员开始关注道路段之间的空间关系。一些两步方法先利用道路网络拓扑建模空间关系，然后使用序列模型进行时间建模。



 **轨迹简化方法：**由于GPS轨迹中存在冗余和噪声，使用路径轨迹代替原始GPS轨迹成为主流。这些方法引入了许多自然语言处理（NLP）技术，如Word2Vec和BERT，因为路径轨迹与自然语言句子具有相似性。

 **自监督学习方法：**为了在无任务的情况下训练轨迹表示模型，设计了多种自监督训练方法。



**Traj2vec:** 将轨迹序列编码为隐藏表示，解码根据隐藏表示重建轨迹序列。适用于需要轨迹整体表示的任务，如轨迹聚类和分类

**NeuTraj、T3S:** 原始GPS轨迹简化为路径轨迹来减少噪声和冗余。使用地图匹配算法将GPS点映射到道路网络上，形成路径轨迹。

**TrajCL:** 通过对比学习的方式，设计多种数据增强策略来生成正负样本对，从而学习到具有判别性的轨迹表示。适用于需要高判别性轨迹表示的任务，如轨迹分类和相似性搜索。



## II 相关工作



## II-A GPS轨迹表示学习

•**时空点序列**: GPS轨迹是包含大量时间和空间信息的时空点序列。

•**不同于一般序列建模**: 与只考虑时间因素的一般序列建模不同, GPS轨迹建模需要同时考虑时间和空间因素。

•**轨迹简化**: 为了更好地对GPS轨迹中的空间属性进行建模, 简化轨迹可以分为两大类: 基于窗口的和基于道路网络的。经过道路网络简化的GPS轨迹称为路线轨迹。

### 轨迹简化方法

- 基于窗口的方法**: **traj2vec**: 使用自定义时间或空间窗口顺序扫描GPS轨迹。序列模型将每个窗口编码为序列中的一个标记。通过滑动窗口捕捉轨迹的局部特征; **t2vec**: 优化窗口扫描策略, 更加关注空间建模
- 基于道路网络的方法**: **TrajCL**: 利用离散化的光栅窗口对原始轨迹进行处理, 通过轨迹拓扑结构的简化, 捕捉轨迹的主要形状和路径; **图自编码器(GAE)**: 利用图自编码器学习节点表示, 能够捕捉到道路网络的拓扑结构信息。

### 捕捉时空依赖性

- NeuTraj**: 利用变分自编码器 (VAE) 和对抗训练, 来捕捉轨迹数据中的非线性结构和复杂分布; **T3S**: 对轨迹进行语义分割, 将轨迹划分为多个语义段; 注意力机制捕捉语义段之间的长程依赖关系

### 自监督学习

- TrajCL**: 引入对比学习并设计多个轨迹数据扩充策略来训练模型, 通过随机采样、噪声添加等技术生成正负样本对, 利用对比损失函数优化模型参数。
- 局限性**: 上述方法过于关注宏观转变而忽略了运动细节。最近的研究表明, 使用原始轨迹有助于对细粒度运动模式进行建模, 从而更好地捕捉移动性。
- 改进方向**: 提出一种分层的编码方法, 捕获来自GPS轨迹的稀疏信息, 并解决原始轨迹中的噪声和冗余问题

## II-B 道路轨迹表示学习

### •路径轨迹作为序列数据:

路径轨迹可以被视为序列数据, 其中每个元素代表一个道路段的访问记录。

•**语义与时空特征:** 路径轨迹不仅包含空间信息, 还反映了用户的出行意图和偏好, 因此需要同时捕捉其语义和时空特征。

### 序列模型

将路径轨迹视为序列数据, 利用自然语言处理中的技术 (如**Word2vec**和**Node2vec**) 来学习路径轨迹的表示。通过捕捉道路段的上下文关系, 有效地表示道路段的语义信息。

### 捕捉时空依赖性

•**Trember**: 对轨迹进行时空特征提取, 形成时空序列数据, 再利用RNN对序列进行建模; **START**: **方法细节**: 轨迹编码器对轨迹进行编码时考虑时间和语义信息, 然后设计两个自监督任务, 如预测轨迹中的下一个道路段或重建被掩盖的时间段等, 通过这些任务训练模型, 使模型能够学习到轨迹的表示。

### 自监督学习

•**Toast**: 先预训练得到初始嵌入, 随机掩盖部分道路段, 训练模型预测被掩盖的道路段; 同时通过轨迹判别任务区分真实轨迹和生成的虚假轨迹, 优化模型参数; **PIM**: 利用对比学习, 以道路网络中最短路径作为正样本, 通过交换正路径中的部分节点生成负样本; **JCRLNT**: 图编码器用于获取道路段的表示, 轨迹编码器用于获取轨迹的表示, 然后通过对比学习的方式, 设计三种对比任务 (如道路段与轨迹的对比、不同轨迹的对比等) 来训练模型, 使模型能够学习到道路段和轨迹的有效表示。



# III 研究方法





# IV JGRM框架

•**JGRM框架**: 目标是通过联合建模GPS轨迹和路径轨迹, 充分利用两者的优势, 克服单一数据源的局限性, 实现全面、准确的轨迹表示

## GPS编码器(GPS Encoder):

- ① **分层设计**: 采用分层的双向门控循环单元 (BiGRU) 来处理GPS轨迹
- ② **子轨迹处理**: 将GPS轨迹分解为多个子轨迹, 并对每个子轨迹进行独立建模
- ③ **道路段嵌入**: 将每个子轨迹的特征嵌入到道路段中, 得到每个道路段的特征表示
- ④ **序列建模**: 通过道路段之间的序列关系优化表示, 捕获上下文信息

## 路径编码器(Route Encoder):

- ① **空间建模**: 图注意力网络 (GAT) 来捕捉道路段之间的空间关系
- ② **时间建模**: 轻量级时间编码器 (TE) 捕捉道路段的访问时间模式
- ③ **自相关性建模**: Transformer对路径轨迹进行自相关性建模。捕捉道路段序列中的长程依赖关系
- ④ **特征融合**: 将空间、时间和自相关性特征进行融合, 得到每个道路段的综合特征表示

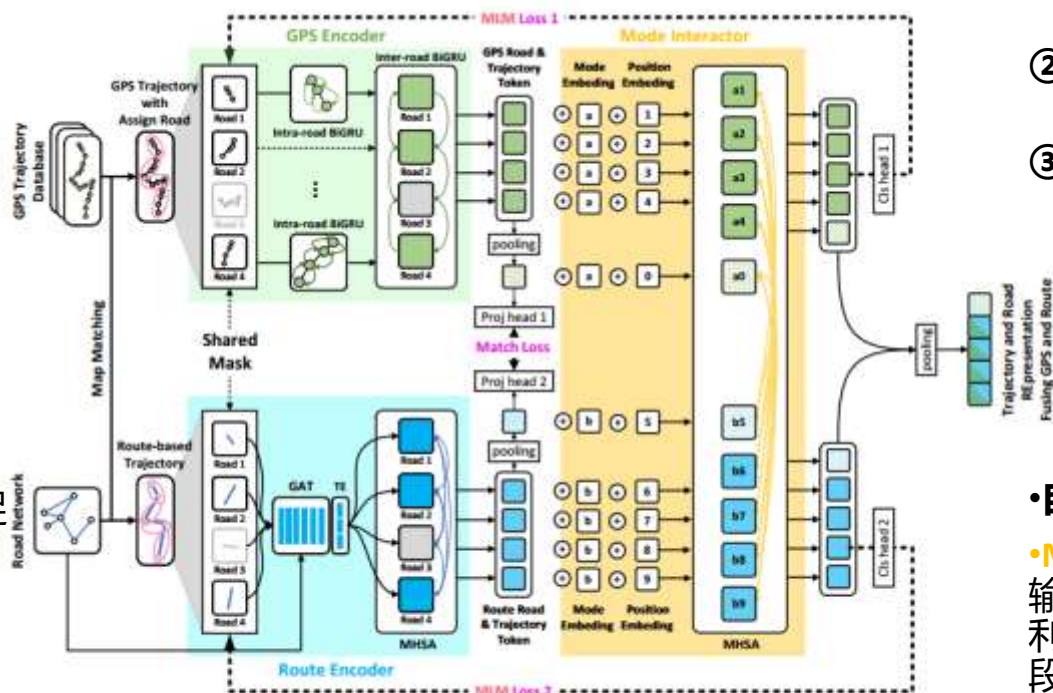


Figure 2: The Framework of JGRM.

## 模态交互器 (Modal Interactor)

- ① **模态嵌入**: 对每个模态的输入进行模态嵌入和位置嵌入, 保留模态身份信息。
- ② **共享Transformer**: 捕捉不同模态之间的相关性, 确保信息的充分融合。
- ③ **表示对齐**: 利用自监督任务如MLM和CMM确保两种模态的表示空间对齐, 从而实现更有效的信息融合。

## •自监督训练任务:

•**MLM (Masked Language Modeling)**: 在轨迹输入编码器之前, 随机掩盖一些道路段, 然后利用模态交互器的输出重建这些被掩盖的道路段, 通过重建误差作为监督信号来训练模型, 以捕捉轨迹本身的时空连续性。

•**CMM (Cross-Modal Matching)**: 利用GPS轨迹和路径轨迹的配对关系, 生成成对损失, 引导两种模态在送入模态交互器之前对齐表示空间, 从而实现视图间信息的有效融合。



## IV-A GPS编码器

### 数据预处理与子轨迹划分：

使用地图匹配算法将原始GPS轨迹映射到道路网络上，创建分配矩阵  
提取每个GPS点的特征，包括**经度、纬度、速度、加速度、角度变化、时间间隔和距离**，形成特征矩阵

### 子轨迹特征提取：

•根据分配矩阵，将特征矩阵组织成子轨迹的特征矩阵。

### 层内BiGRU (Intra-Road BiGRU)：

将每个子轨迹的特征矩阵输入到层内BiGRU中进行编码。BiGRU分别从正向和反向两个方向处理序列数据，得到每个子轨迹的隐藏表示

### 层间BiGRU (Inter-Road BiGRU)：

将层内BiGRU得到的隐藏表示作为输入，送入层间BiGRU中进行进一步处理。  
层间BiGRU通过考虑子轨迹之间的顺序关系，对道路段表示进行精炼

### 道路段表示与轨迹表示的生成：

- 将层间BiGRU的前向和后向隐藏表示进行拼接，得到最终的道路段表示
- 对所有道路段的表示进行平均池化操作，得到轨迹在GPS视图下的表示

## 地图匹配算法

基于道路网络的拓扑结构和几何形状，将原始GPS点分配到最可能的道路段上。如GPS点与道路的距离、道路的方向、道路的连通性以及车辆的运动学特征（如速度和方向）等，以确定最佳匹配。

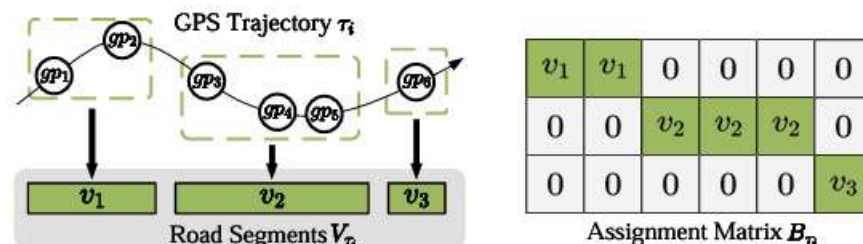


Figure 3: An example of an assignment matrix.



## IV-B 路径编码器

### 特征嵌入:



- 路径轨迹的特征包括道路ID、时间间隔、分钟索引（0-1439）和星期索引（0-6），形成特征矩阵
- 将道路ID嵌入为密集向量，时间特征（分钟索引和星期几索引）作为离散值嵌入，实际行驶时间作为连续值嵌入

### 图注意力网络(GAT):



- 使用GAT更新道路段的嵌入表示，以捕捉道路网络的拓扑结构。GAT通过消息传递机制更新道路段的表示：

### 时间编码器(TE):



时间编码器用于捕捉时间特征，包括分钟索引和星期几索引。时间特征被嵌入为离散值，实际行驶时间被嵌入为连续值

### Transformer建模自相关性:



Transformer通过多头自注意力机制和前馈神经网络对序列进行建模，确保每个道路段的表示能够反映其在整个轨迹中的上下文信息。

# IV 实验分析

# IV 实验分析

## ■ 实验设置

- **Chengdu、Xian:** 包含GPS轨迹、路线轨迹和道路网络
- **GPS轨迹来自滴滴出行的公共数据集**
- **道路网络: OSMnx:** 从OpenStreetMap下载街道网络数据, 包括步行、驾驶、骑行等多种类型的网络。此外, 还可以获取城市设施、建筑物轮廓、公共交通站点等其他地理空间特征
- **数据预处理:**
  - 使用地图匹配算法将GPS轨迹映射到道路网络上
  - 过滤掉未被轨迹覆盖的道路段
  - 去除轨迹长度过短的轨迹
- **时间跨度15天:** 前13天训练集、第14天验证集、第15天测试集

## ■ 四个下游任务

- 实验设计了四个下游任务, 包括两个道路段级别的任务(道路分类和道路速度推断)和两个轨迹级别的任务(旅行时间估计和Top-k相似轨迹查询)。
- 评估指标针对这些任务, 分别采用了Micro-F1、Macro-F1、MAE、RMSE、MR、HR@10等评估指标来衡量模型的表现。

Table 3: Details of the Datasets

Datasets	Chengdu	Xi'an
Region Sizes ( $km^2$ )	68.26	65.62
# Nodes	6450	4996
# Edges	16398	11864
# Trajectories	2140129	1289037
Avg. Trajectory Length ( $m$ )	2857.81	2976.52
Avg. Road Travel Speed ( $m/s$ )	11.35	9.65
Avg. Trajectory Travel Time ( $s$ )	436.12	516.24
Time span	2018/11/01 - 2018/11/15	

# IV 实验分析

- 道路段级别任务
- 轨迹级别任务

## ■ 性能比较

- 随机初始化: 直接为每个道路段生成随机的向量作为其表示, 不利用任何轨迹数据或道路网络结构信息。

## ■ 基于图的轨迹表示学习:

1. Word2vec
2. Node2vec
3. GAE (Graph Auto-Encoder)

## ■ 基于GPS的轨迹表示学习:

1. Traj2vec

## ■ 基于道路路径的轨迹表示学习:

1. Toast
2. PIM
3. Trember
4. START
5. JCRLNT

Table 1: Model comparison on four downstream tasks in Chengdu.

	Road Classification		Road Speed Inference		Travel Time Estimation		Top-k Similar Trajectory Query		
	Mi-F1	Ma-F1	MAE	RMSE	MAE	RMSE	MR	HR@10	No Hit
Embedding	0.3853	0.2757	3.561	4.6437	102.592	132.4559	9.4693	0.85	0
Word2vec	0.5514	0.5137	3.5004	4.5424	87.1612 <sup>‡</sup>	115.6605 <sup>‡</sup>	12.4355	0.7998	0
Node2vec	0.408	0.364	3.5761	4.6623	88.1243	117.3834	4.103 <sup>‡</sup>	0.9127 <sup>‡</sup>	0
GAE	0.4373	0.3805	3.287 <sup>‡</sup>	4.2134 <sup>‡</sup>	90.2352	122.9764	4.4584 <sup>‡</sup>	0.9067 <sup>‡</sup>	0
Traj2vec	0.4828	0.399	2.856 <sup>‡</sup>	3.81 <sup>‡</sup>	99.0706	128.4441	67.5899	0.55	839.2
Toast	0.6276 <sup>‡</sup>	0.6195 <sup>‡</sup>	3.3201	4.3777	86.0053 <sup>‡</sup>	114.2109 <sup>‡</sup>	5.9169	0.8696	0
PIM	0.4618	0.4457	3.4841	4.5737	87.6526	116.533	5.109	0.8902	0
Trember	0.611 <sup>‡</sup>	0.6059 <sup>‡</sup>	3.3955	4.447	90.9035	119.0926	17.9627	0.7427	0.1
START	0.409	0.3366	3.5269	4.6084	89.7182	117.9891	6.9448	0.909	30.7
JCRLNT	0.5169	0.466	3.441	4.5016	100.1113	129.591	20.0152	0.7323	0.6
JGRM	<b>0.7198</b>	<b>0.7228</b>	<b>2.5783</b>	<b>3.5452</b>	<b>83.3306</b>	<b>110.7224</b>	<b>2.2111</b>	<b>0.9492</b>	0
JGRM*	<b>0.8067*</b>	<b>0.8111*</b>	<b>2.3162*</b>	<b>3.2953*</b>	<b>80.4002*</b>	<b>108.0134*</b>	<b>1.1363*</b>	<b>0.9735*</b>	0
improvement	14.69%	16.67%	10.77%	7.47%	3.21%	3.15%	85.56%	4%	/
improvement*	28.54%	30.93%	23.31%	15.62%	6.97%	5.74%	261.08%	6.66%	/

Table 4: Model comparison on four downstream tasks in Xi'an.

	Road Classification		Road Speed Inference		Travel Time Estimation		Top-k Similar Trajectory Query		
	Mi-F1	Ma-F1	MAE	RMSE	MAE	RMSE	MR	HR@10	No Hit
Embedding	0.4382	0.3003	3.2619	4.1949	104.5929	137.0655	4.0946	0.9031	0
Word2vec	0.5962	0.5559	3.2242	4.1103	92.9827	129.9678	5.795	0.8617	0
Node2vec	0.4283	0.3827	3.2945	4.236	89.6014 <sup>‡</sup>	122.2406 <sup>‡</sup>	3.1167 <sup>‡</sup>	0.923 <sup>‡</sup>	0
GAE	0.462	0.436	3.2496	4.1794	90.2352 <sup>‡</sup>	122.9764 <sup>‡</sup>	3.5626	0.9141	0
Traj2vec	0.5658	0.4195	2.7798 <sup>‡</sup>	3.6768 <sup>‡</sup>	107.8969	144.248	51.6097	0.6221	361.5
Toast	0.7055 <sup>‡</sup>	0.6606 <sup>‡</sup>	3.1145	4.0025	92.9093	129.3365	5.0072	0.869	0
PIM	0.512	0.4671	3.2367	4.1845	91.0666	123.6043	4.243	0.8947	0
Trember	0.6627 <sup>‡</sup>	0.6212 <sup>‡</sup>	3.2052	4.1269	98.8188	134.7582	9.5947	0.8084	0
START	0.4557	0.3298	3.2211	4.1331	105.8333	138.6432	<b>2.5158</b>	0.9283 <sup>‡</sup>	6.7
JCRLNT	0.609	0.5179	3.1651 <sup>‡</sup>	4.0864 <sup>‡</sup>	100.8771	133.8522	13.4306	0.7659	0
JGRM	<b>0.7823</b>	<b>0.7703</b>	<b>2.6494</b>	<b>3.5818</b>	<b>87.166</b>	<b>119.2541</b>	2.7714 <sup>‡</sup>	<b>0.9294</b>	0
JGRM*	<b>0.8758*</b>	<b>0.8698*</b>	<b>2.2029*</b>	<b>3.1765*</b>	<b>86.2855*</b>	<b>118.9211*</b>	<b>1.2983*</b>	<b>0.9682*</b>	0
improvement	10.89%	16.61%	4.92%	2.65%	2.79%	2.5%	/	0.12%	/
improvement*	24.14%	31.67%	26.19%	15.75%	3.84%	2.79%	93.78%	4.3%	/



## IV 实验分析

- 道路段级别任务
- 轨迹级别任务

JGRM道路段级任务上的表现要好得多

有效的道路段建模可以显著提高轨迹表示学习的性能。

处理轨迹序列的模型：  
Toast、PIM、Trember在  
轨迹级任务中往往表现得更好

轨迹中建模时空相关性是必要的。

基于GPS的表示学习方法  
traj2vec表现不佳

GPS轨迹中的噪声和冗余造成的

基于路径表示的学习方法在  
序列级任务上发挥更好

这表明道路段之间的拓扑结构对于轨迹表示是重要的

Table 1: Model comparison on four downstream tasks in Chengdu.

	Road Classification		Road Speed Inference		Travel Time Estimation		Top-k Similar Trajectory Query		
	Mi-F1	Ma-F1	MAE	RMSE	MAE	RMSE	MR	HR@10	No Hit
Embedding	0.3853	0.2757	3.561	4.6437	102.592	132.4559	9.4693	0.85	0
Word2vec	0.5514	0.5137	3.5004	4.5424	87.1612 <sup>‡</sup>	115.6605 <sup>‡</sup>	12.4355	0.7998	0
Node2vec	0.408	0.364	3.5761	4.6623	88.1243	117.3834	4.103 <sup>‡</sup>	0.9127 <sup>‡</sup>	0
GAE	0.4373	0.3805	3.287 <sup>‡</sup>	4.2134 <sup>‡</sup>	90.2352	122.9764	4.4584 <sup>‡</sup>	0.9067 <sup>‡</sup>	0
Traj2vec	0.4828	0.399	2.856 <sup>‡</sup>	3.81 <sup>‡</sup>	99.0706	128.4441	67.5899	0.55	839.2
Toast	0.6276 <sup>‡</sup>	0.6195 <sup>‡</sup>	3.3201	4.3777	86.0053 <sup>‡</sup>	114.2109 <sup>‡</sup>	5.9169	0.8696	0
PIM	0.4618	0.4457	3.4841	4.5737	87.6526	116.533	5.109	0.8902	0
Trember	0.611 <sup>‡</sup>	0.6059 <sup>‡</sup>	3.3955	4.447	90.9035	119.0926	17.9627	0.7427	0.1
START	0.409	0.3366	3.5269	4.6084	89.7182	117.9891	6.9448	0.909	30.7
JCRLNT	0.5169	0.466	3.441	4.5016	100.1113	129.591	20.0152	0.7323	0.6
JGRM	<b>0.7198</b>	<b>0.7228</b>	<b>2.5783</b>	<b>3.5452</b>	<b>83.3306</b>	<b>110.7224</b>	<b>2.2111</b>	<b>0.9492</b>	0
JGRM*	<b>0.8067*</b>	<b>0.8111*</b>	<b>2.3162*</b>	<b>3.2953*</b>	<b>80.4002*</b>	<b>108.0134*</b>	<b>1.1363*</b>	<b>0.9735*</b>	0
improvement	14.69%	16.67%	10.77%	7.47%	3.21%	3.15%	85.56%	4%	/
improvement*	28.54%	30.93%	23.31%	15.62%	6.97%	5.74%	261.08%	6.66%	/

Table 4: Model comparison on four downstream tasks in Xi'an.

	Road Classification		Road Speed Inference		Travel Time Estimation		Top-k Similar Trajectory Query		
	Mi-F1	Ma-F1	MAE	RMSE	MAE	RMSE	MR	HR@10	No Hit
Embedding	0.4382	0.3003	3.2619	4.1949	104.5929	137.0655	4.0946	0.9031	0
Word2vec	0.5962	0.5559	3.2242	4.1103	92.9827	129.9678	5.795	0.8617	0
Node2vec	0.4283	0.3827	3.2945	4.236	89.6014 <sup>‡</sup>	122.2406 <sup>‡</sup>	3.1167 <sup>‡</sup>	0.923 <sup>‡</sup>	0
GAE	0.462	0.436	3.2496	4.1794	90.2352 <sup>‡</sup>	122.9764 <sup>‡</sup>	3.5626	0.9141	0
Traj2vec	0.5658	0.4195	2.7798 <sup>‡</sup>	3.6768 <sup>‡</sup>	107.8969	144.248	51.6097	0.6221	361.5
Toast	0.7055 <sup>‡</sup>	0.6606 <sup>‡</sup>	3.1145	4.0025	92.9093	129.3365	5.0072	0.869	0
PIM	0.512	0.4671	3.2367	4.1845	91.0666	123.6043	4.243	0.8947	0
Trember	0.6627 <sup>‡</sup>	0.6212 <sup>‡</sup>	3.2052	4.1269	98.8188	134.7582	9.5947	0.8084	0
START	0.4557	0.3298	3.2211	4.1331	105.8333	138.6432	2.5158	0.9283 <sup>‡</sup>	6.7
JCRLNT	0.609	0.5179	3.1651 <sup>‡</sup>	4.0864 <sup>‡</sup>	100.8771	133.8522	13.4306	0.7659	0
JGRM	<b>0.7823</b>	<b>0.7703</b>	<b>2.6494</b>	<b>3.5818</b>	<b>87.166</b>	<b>119.2541</b>	<b>2.7714<sup>‡</sup></b>	<b>0.9294</b>	0
JGRM*	<b>0.8758*</b>	<b>0.8698*</b>	<b>2.2029*</b>	<b>3.1765*</b>	<b>86.2855*</b>	<b>118.9211*</b>	<b>1.2983*</b>	<b>0.9682*</b>	0
improvement	10.89%	16.61%	4.92%	2.65%	2.79%	2.5%	/	0.12%	/
improvement*	24.14%	31.67%	26.19%	15.75%	3.84%	2.79%	93.78%	4.3%	/



# IV 实验分析

## 消融实验

Table 2: Ablation experiment on four downstream tasks in Chengdu.

	Road Classification		Road Speed Inference		Travel Time Estimation		Top-k Similar Trajectory Query		
	Mi-F1	Ma-F1	MAE	RMSE	MAE	RMSE	MR	HR@10	No Hit
JGRM	0.7198	0.7228	2.5783	3.5452	83.3306	110.7224	2.2111	0.9492	0
w/o MLM Loss	0.5233	0.4804	3.4752	4.5521	122.7088	152.9668	26.4418	0.0085	4725.8
w/o Match Loss	0.7178	0.7232 ↑	2.6075	3.5947	82.5453 ↑	110.2262 ↑	2.3396	0.9441	0
w/o GPS Branch	0.6245	0.6206	3.2008	4.2258	83.6647	111.4075	1.6037 ↑	0.963 ↑	0
w/o Route Branch	0.6122	0.5929	2.8302	3.7668	95.2015	124.4988	9.2601	0.8381	0
w/o Time Info	0.7331 ↑	0.7361 ↑	2.6225	3.5866	84.1749	111.6983	5.6927	0.8745	0
w/o Mode Interactor	0.6043	0.5859	2.7381	3.7303	82.9407 ↑	110.4866 ↑	1.4601 ↑	0.965 ↑	0
w/o GAT	0.7173	0.7225	2.706	3.654	82.2657 ↑	110.038 ↑	1.1554 ↑	0.9732 ↑	0
w/o Mode Emb	0.7161	0.7222	2.7439	3.6944	83.8222	111.5119	2.535	0.9417	0

Table 5: Ablation experiment on four downstream tasks in Xi'an.

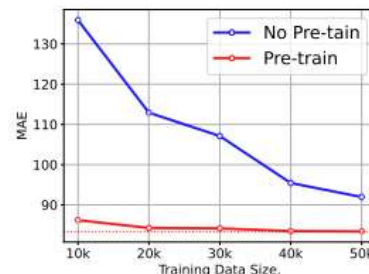
	Road Classification		Road Speed Inference		Travel Time Estimation		Top-k Similar Trajectory Query		
	Mi-F1	Ma-F1	MAE	RMSE	MAE	RMSE	MR	HR@10	No Hit
JGRM	0.7823	0.7703	2.6494	3.5818	87.166	119.2541	2.7714	0.9294	0
w/o MLM Loss	0.5327	0.4128	3.2402	4.1623	115.9861	148.8677	75.0366	0.0768	3855.2
w/o Match Loss	0.7793	0.7666	2.5667 ↑	3.5338 ↑	87.3213	119.262	2.7729	0.9319	0
w/o GPS Branch	0.7003	0.6869	2.7983	3.7388	87.1901	119.3732	2.2322 ↑	0.9441 ↑	0
w/o Route Branch	0.6248	0.5717	2.7472	3.5753	98.0748	131.2151	5.7801	0.8663	0
w/o Time Info	0.7745	0.7601	2.5816 ↑	3.5254 ↑	87.5762	119.8214	5.65	0.8655	0
w/o Mode Interactor	0.6268	0.5757	2.8074	3.7472	87.2887	119.3806	2.0412	0.9492 ↑	0
w/o GAT	0.7987 ↑	0.7846 ↑	2.6676	3.5982	87.2087	118.8381 ↑	1.7644 ↑	0.956 ↑	0
w/o Mode Emb	0.7802	0.7691	2.4292 ↑	3.3746 ↑	87.0462 ↑	118.757 ↑	3.1417	0.9245	0

## 模型容量与性能关系

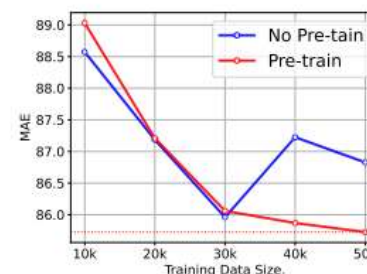
- 模型容量的影响：随着训练数据量的增加，JGRM的性能持续提升。这表明JGRM具有较大的模型容量，能够随着训练数据的增加不断提升性能。
- 模型潜力：JGRM展现出作为交通基础设施大型模型的潜力。其在大规模数据上的性能提升表明，模型能够有效利用更多的数据来学习更复杂的模式和特征。

## 预训练模型效果研究

- 数据需求减少：在训练数据较少时能够显著减少训练所需的数据量，展现出丰富的先验
- 防止过拟合：随着训练数据的增加，模型性能持续提升。在西安数据集上，随着训练数据的增加，预训练模型的MAE逐渐降低，而未预训练模型的MAE在数据量较大时出现波动。

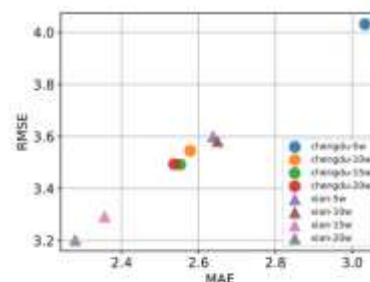


(a) MAE in Chengdu.

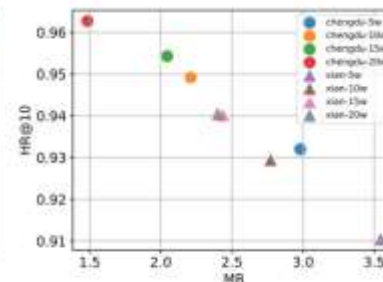


(b) MAE in Xi'an.

Figure 4: Effect of pre-training in travel time estimation.



(a) Speed Inference.



(b) Similar Trajectory Search.

Figure 5: Model Capacity.

# IV 实验分析

## 参数敏感性

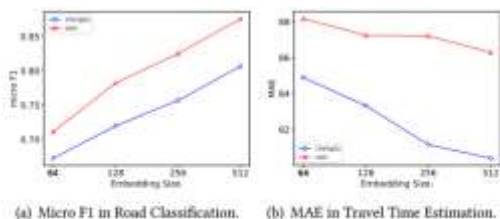


Figure 6: Different # of Embedding Sizes.

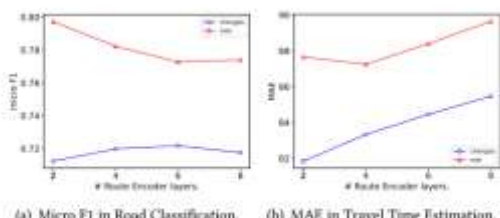


Figure 7: Different # of Route Layers.

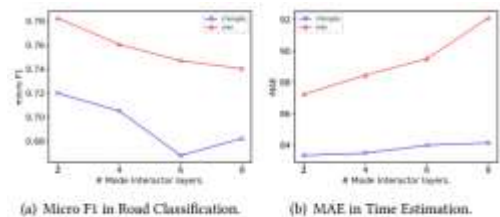


Figure 8: Different # of Mode Interact Layers.

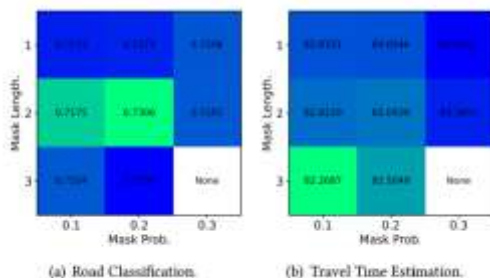


Figure 9: Different # of Mask Settings.

### • 嵌入大小 (Embedding Size) :

- 较大的嵌入大小能提升模型性能, 最佳性能在嵌入大小为1024时取得
- 这表明轨迹数据存在复杂模式, 需要高维空间表示。

### • 路径编码器层数 (Route Encoder Layers) :

- 参数在不同城市的实验结果有差异, 但总体在层数为2时取得最佳结果
- 这可能与不同城市轨迹的复杂程度不同有关。

### • 模态交互器层数 (Mode Interactor Layers) :

- 2层的模态交互器性能最佳, 能有效融合GPS和路径信息

### • 掩码长度和概率 (Mask Length and Probability) :

- 当掩码长度约为轨迹的40%时, 模型性能最佳
- 且在掩码数量相同的情况下, 较长的掩码长度能提升模型在轨迹级别任务上的性能。

## 模态交互器对道路段和轨迹表示的影响

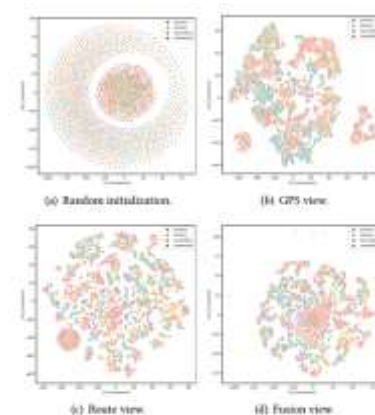
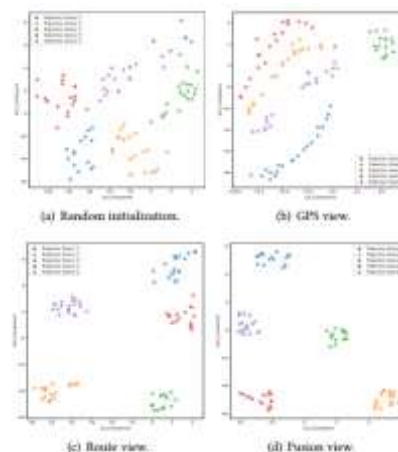


Figure 10: Road Segment Representation Space.

### 轨迹表示:

- 从成都数据集中随机选取五个互不相似的轨迹作为查询轨迹, 分别使用JGRM和Node2vec方法获取轨迹表示, 并找出每个查询轨迹的前20个相似轨迹
- JGRM的两个编码器能有效编码轨迹, 模态交互器有助于对齐轨迹的表示空间, 使轨迹表示更加准确和鲁棒。

### 道路段表示:

- 通过可视化不同模态 (随机初始化、GPS视图、路径视图、融合视图) 下的道路段表示, 发现GPS视图和路径视图均能有效建模道路段特征, 而模态交互器进一步融合两种模态的信息, 提升了道路段表示的质量。

## IV 实验分析

### ■ 模型迁移能力研究

- 零样本适应 (Zero-shot Adaptation) : 将模型在源城市训练得到的参数直接应用于目标城市, 不进行任何调整。
  - 在路段级别任务上能够达到在目标城市直接训练模型性能的约90%; 在轨迹级别任务上, 零样本适应的性能较差。
- 少样本微调 (Few-shot Fine-tuning) : 使用目标城市的少量数据对在源城市训练的模型进行微调。
  - 少样本微调能够显著提升模型在轨迹级别任务上的性能。
  - 路段级别任务的性能变化: 少样本微调虽然提升了轨迹级别任务的性能, 但在某些情况下可能会导致路段级别任务的性能略有下降。这可能是因为微调过程中, 模型过于关注目标城市中更新的路段表示, 而忽略了一些未被充分覆盖的路段。

**Table 6: Model transferability across two cities.**

		<b>0.7823</b>	<b>0.7703</b>	<b>87.166</b>	<b>119.2541</b>
		Road Classification		Travel Time Estimation	
		Mi-F1	Ma-F1	MAE	RMSE
Zero Shot Adaptation	C→X	0.7252	0.6873	109.206	141.6533
	X→C	0.7295	0.6916	106.5079	139.2584
Few Shot Finetune	C→X	0.6712	0.6662	105.2994	134.9308
	X→C	0.6802	0.6779	99.1057	128.7578

C and X in the table are abbreviations for Chengdu and Xi'an.

谢谢！

