北京航空航天大学计算机学院
School of Computer Science and Engineering, Beihang University

自监督轨迹表征学习方法研究

姜佳伟
SY2106210

目录

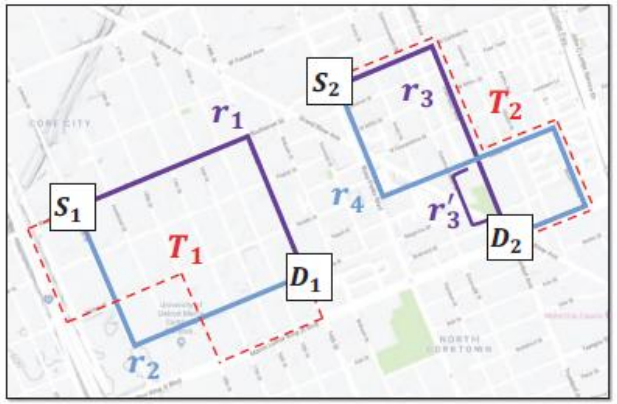
CONTENTS

目录

CONTENTS

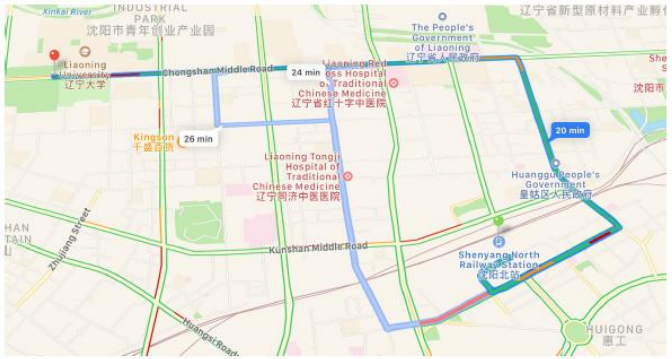
背景：轨迹数据分析与管理

- 随着GPS设备的快速发展，城市中可以收集到大量的**轨迹数据。**
- **轨迹：描述物体的移动路线的一系列位置构成的时间序列。**
- 轨迹数据的分析与管理是工程界的热门话题，包括若干典型的任务和应用。
- 传统的轨迹数据分析研究需要**大量的特征工程**并且**针对特定任务设计**不同的模型，这使得它们难以迁移到不同的应用中去。



轨迹异常检测

轨迹聚类



出行时间预测

背景：轨迹数据分析与管理

□ 传统轨迹数据挖掘

- 严重依赖领域专家的先验知识和经验，设计**人工特征工程**，代价较高。
- 需要为不同的任务开发独特的模型，需要**大量的标记数据**。
- 特定任务设计的模型**难以迁移到其他任务**当中去。

□ 轨迹表征学习

- 旨在将原始轨迹转化为**通用的低维表征向量**。
- 可以**应用于多样的下游任务**，而不是局限于某个特定任务。

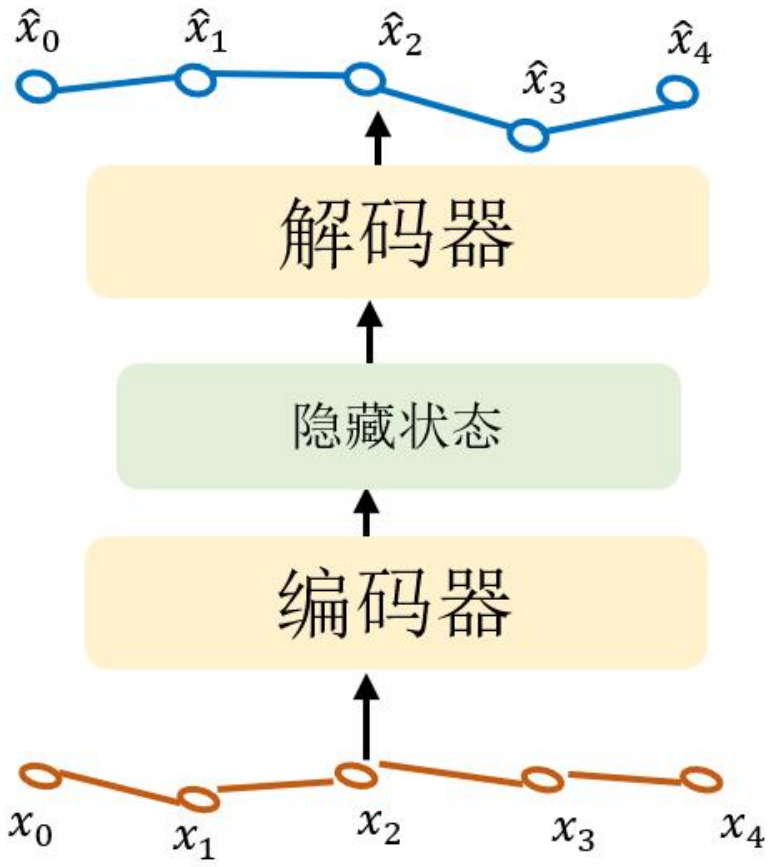
目录

CONTENTS

各领域自监督学习方法

- 自监督的方法主要包括生成式、预测式和对比式方法
  - 生成式：编码器将输入x编码成z，解码器从z重构x，e.g., AutoEncoder
  - 预测式：可以看做是生成式，根据输入x构造标签，从z预测这个标签，e.g., MLM
  - 对比式：构造正负样本，使学习到的表征保持原始数据的相似/不相似关系
- 自监督学习方法
  - CV：Deep InfoMax，SimCLR，MoCo，SwAV…
  - NLP：GPT，BERT，ALBERT，TSDAE，ConSERT，SimSCE…
  - 图：GAE，GVAE，DGI，InfoGraph…
  - 时间序列：CPC，TS-TCC，TS2Vec，CoST…

基于轨迹重构的轨迹表征学习



轨迹重构

- 描述物体的移动路线的一系列**位置构成的时间序列**
- 骨干网络：通用的序列建模的模型，循环神经网络LSTM、Transformer等。
- 使用序列到序列（Seq2Seq）模型，基于轨迹的重构任务生成轨迹表征。

不足

- 仅把轨迹看做当做普通的序列数据（空间轨迹）
- 忽略了其中的**时间规律**

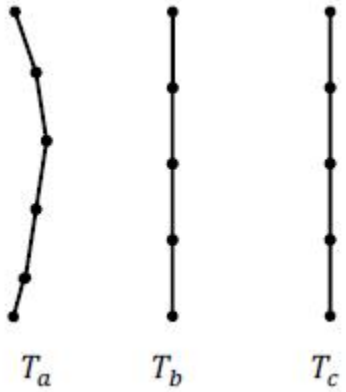
针对特定任务的轨迹表征学习

轨迹相似度计算

- 传统做法：动态时间规整距离 (DTW)， 最长公共子序列距离 (LCSS) 等
- 假设轨迹长度平均为L，则传统轨迹距离计算的时间复杂度一般是平方级别的复杂度，即$\boldsymbol{O(L^2)}$
- 表征做法：学习轨迹表征，以表征向量的距离代表轨迹的距离
- 假设轨迹表征向量的长度为d，则时间复杂度一般为线性复杂度，即$\boldsymbol{O(d)}$
- **针对特定任务的训练：**设轨迹为$\mathcal{T}_i$和$\mathcal{T}_j$，模型学习一个函数$f(\mathcal{T}_i,\mathcal{T}_j)$，给定某个距离指标$d$，模型的目标即使得$|f(\mathcal{T}_i,\mathcal{T}_j)-d(\mathcal{T}_i,\mathcal{T}_j)|$最小

不足

- **不适合跨任务的迁移**

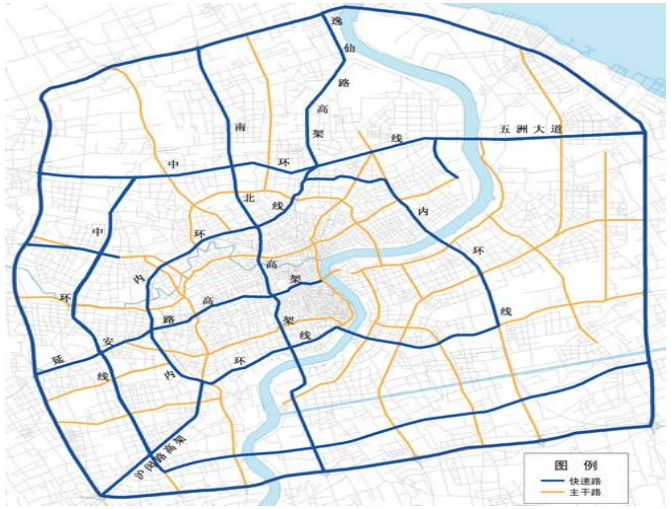


$$T_a = [[1.0, 5.0], [1.4, 4.2], [1.6, 3.3], [1.4, 2.4], [1.2, 1.5], [1.0, 1.0]]$$
$$T_b = [[4.0, 5.0], [4.0, 4.0], [4.0, 3.0], [4.0, 2.0], [4.0, 1.0]]$$
$$T_c = [[7.0, 5.0], [7.0, 4.0], [7.0, 3.0], [7.0, 2.0], [7.0, 1.0]]$$

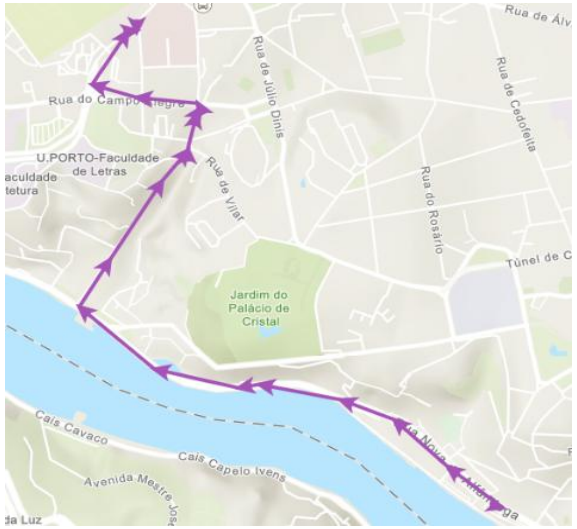
$$d_H(T_b, T_a) = 3.0 == d_H(T_b, T_c) = 3.0$$
$$d_D(T_b, T_a) = 16.5 > d_D(T_b, T_c) = 15.0$$

基于两阶段法的通用轨迹表征学习

路网



轨迹



口 两阶段学习

- 车辆的移动受到路网结构的制约，车辆的轨迹可以看做一系列路段构成的时间序列
- 第一阶段中，采用图表示学习模型来学习路段的表示向量
- 第二阶段中，采用具有自监督任务的序列学习模型来将同一轨迹中的路段表示向量转换为轨迹的表示向量

口 优势

- 把城市路网的静态信息纳入到模型学习的轨迹表征中

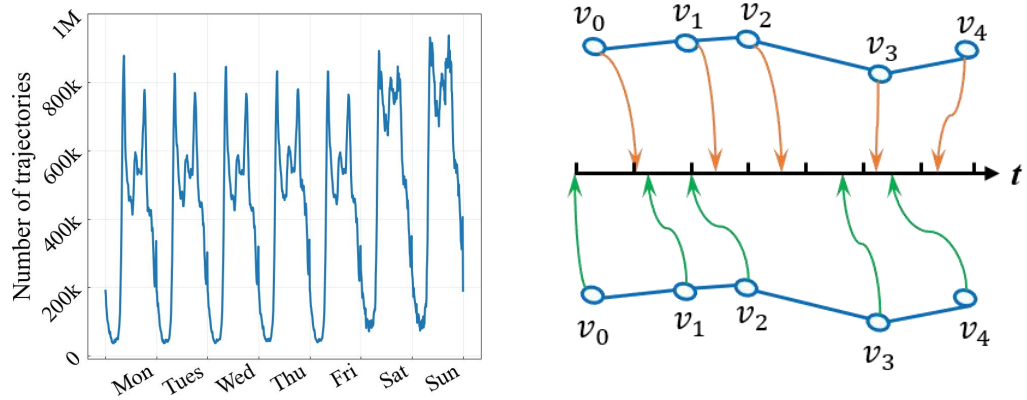
口 不足

- 仅把轨迹看做当做普通的序列数据，忽略了**时间信息**
- 仅考虑静态路网的空间结构，忽略了**旅行语义**

现有方法的不足

- **旅行语义**：轨迹中蕴含着人类的移动模式，过去的工作仅考虑了静态路网，而忽略了动态的道路访问频率等必要信息。
- **时间规律**：一方面轨迹受到城市交通的周期性时间模式的影响，另一方面轨迹采样点之间的时间间隔是不规律的，取决于道路动态的拥堵程度。
- **预训练任务**：过去的工作采用的简单轨迹重构以及轨迹掩码预测没有充分考虑轨迹中的时空特性。

旅行语义



时间规律

目录

CONTENTS

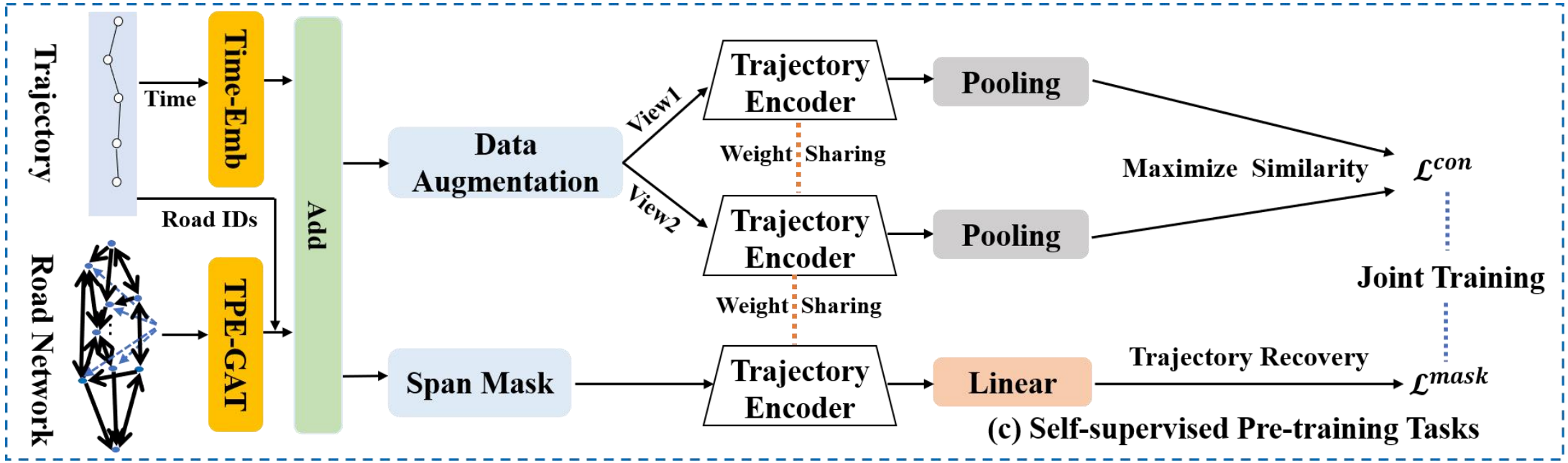
问题定义

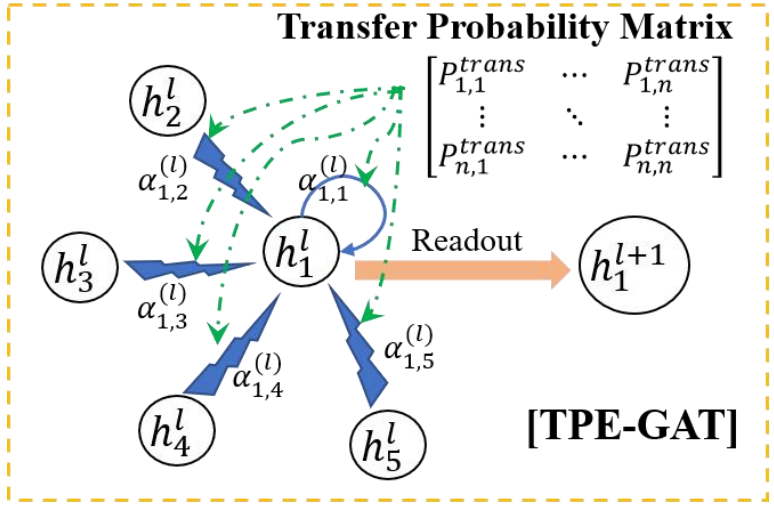
- **路网** 道路网络构成的有向图，节点是路段，边是路段之间的连接
- **GPS轨迹** GPS点的序列，包括每一个点的经度、纬度和时间戳
- **路网限制的轨迹** 路段序列和对应的访问时间戳，由GPS轨迹经过路网匹配后得到
- **轨迹表示学习**

轨迹表示学习的目的是将轨迹转换成通用的低维度的表征向量，以应用于多样的下游任务中。

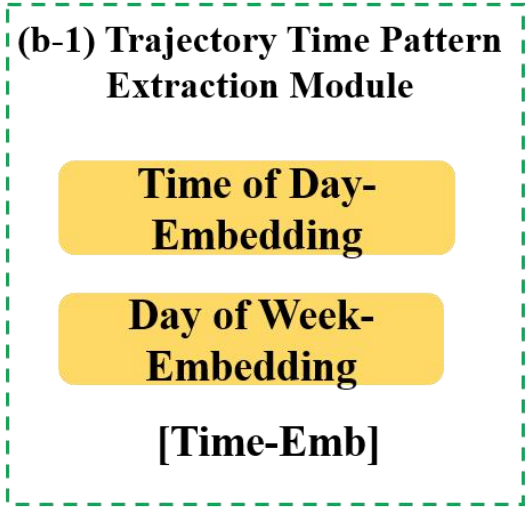
本研究针对路网限制的轨迹。

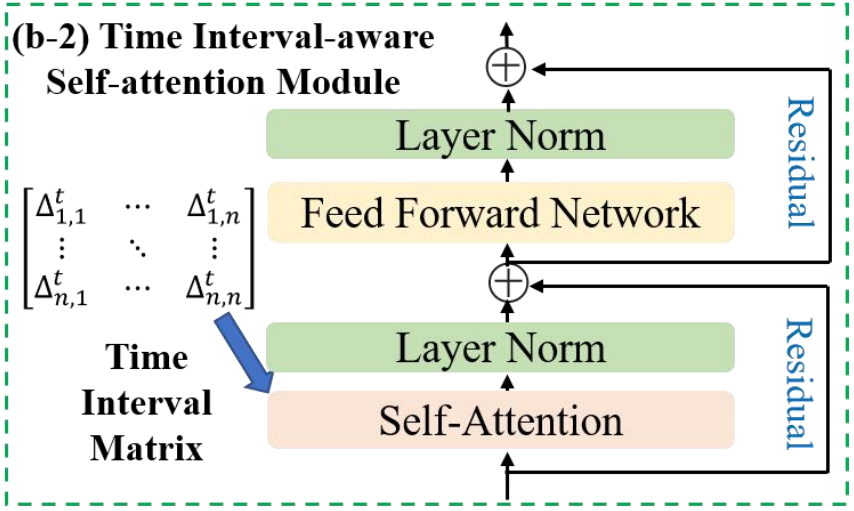
START: 两阶段轨迹表征自监督模型





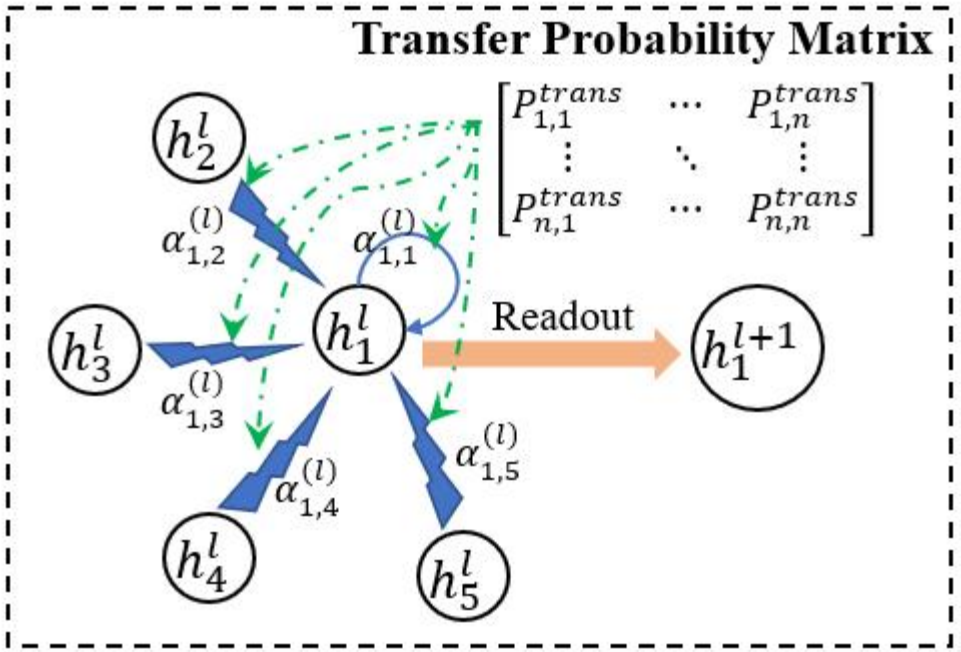
(a) Trajectory Pattern-Enhanced GAT [TPE-GAT]





(b) Time-aware Trajectory Encoder [TAT-Enc]

START: 轨迹模型增强的图神经网络模型（阶段 1）



(a) Trajectory Pattern-Enhanced GAT

- 聚合道路必要特征，得到初始路段向量h_i：

6类特征：道路类别、长度、限速、车道数、入度、出度

- 融合路段转移概率的注意力分数计算：

结合道路i和j的基本特征h_i和h_j以及i和j之间的转移概率p_{ij}^{trans}，计算注意力分数e_{ij}和归一化后的α_{ij}。

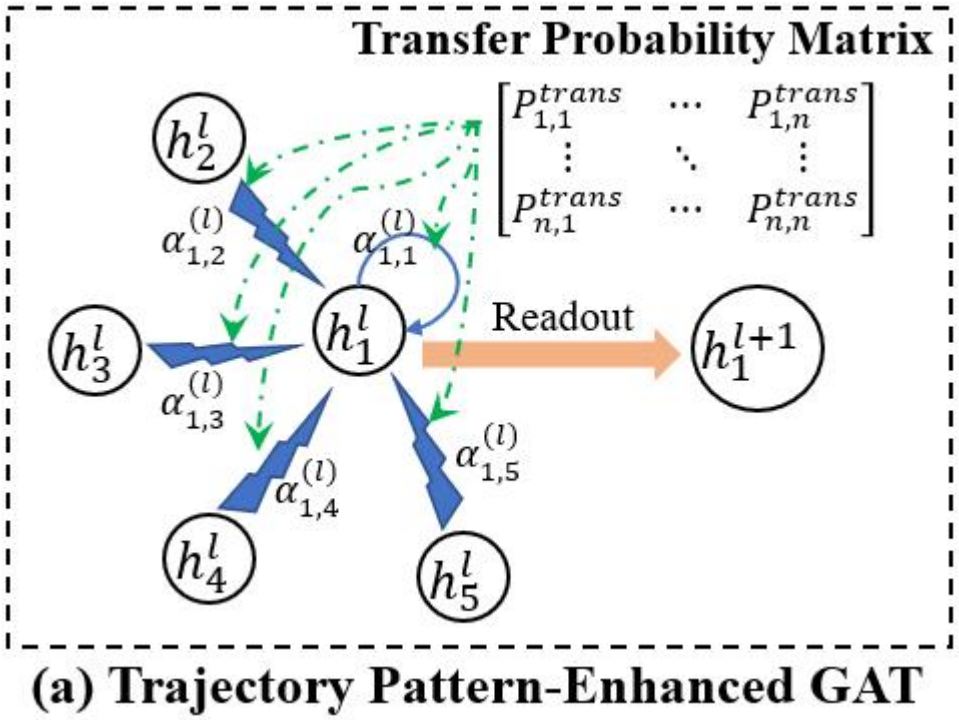
$$e_{ij} = (h_i W_1 + h_j W_2 + p_{ij}^{trans} W_3) W_4^T,$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(e_{ik}))},$$

- 路段之间转移概率的计算：

$$p_{ij}^{trans} = \text{count}(v_i \rightarrow v_j) / \text{count}(v_i),$$

START: 轨迹模型增强的图神经网络模型（阶段 1）



聚合邻居路段的表征：

根据归一化的注意力分数α_{ij}加权聚合邻居路段的表征，

得到更新后的道路表征$\tilde{h}_i$

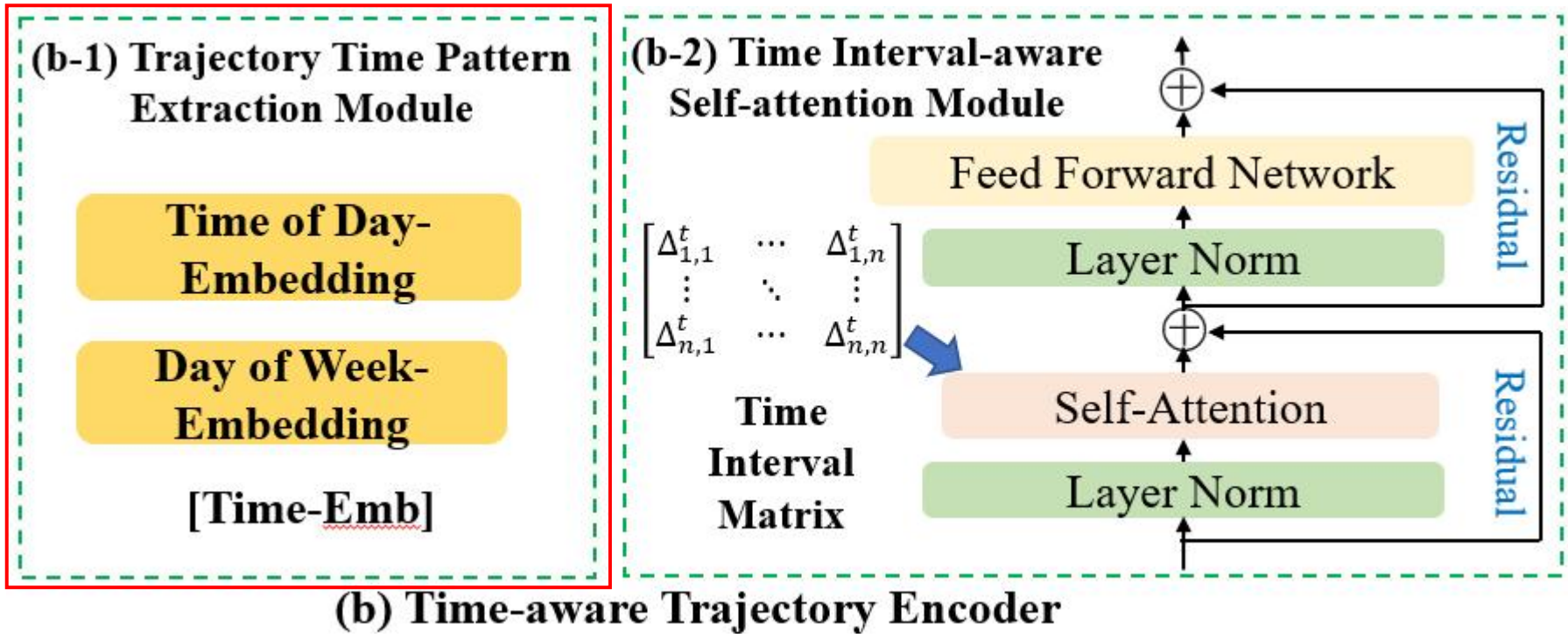
$$\tilde{\boldsymbol{h}}_i^{(l+1)} = \mathrm{ELU}\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\boldsymbol{h}_j^{(l)}\boldsymbol{W}_5\right),$$

多头注意力：

引入多头注意力机制，平稳训练过程，引入多方面的信息

$$\boldsymbol{h}_i^{(l+1)} = \Big\|_{k=1}^{H_1} \mathrm{ELU}\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^{(k)}\boldsymbol{h}_j^{(l)}\boldsymbol{W}_5^{(k)}\right),$$

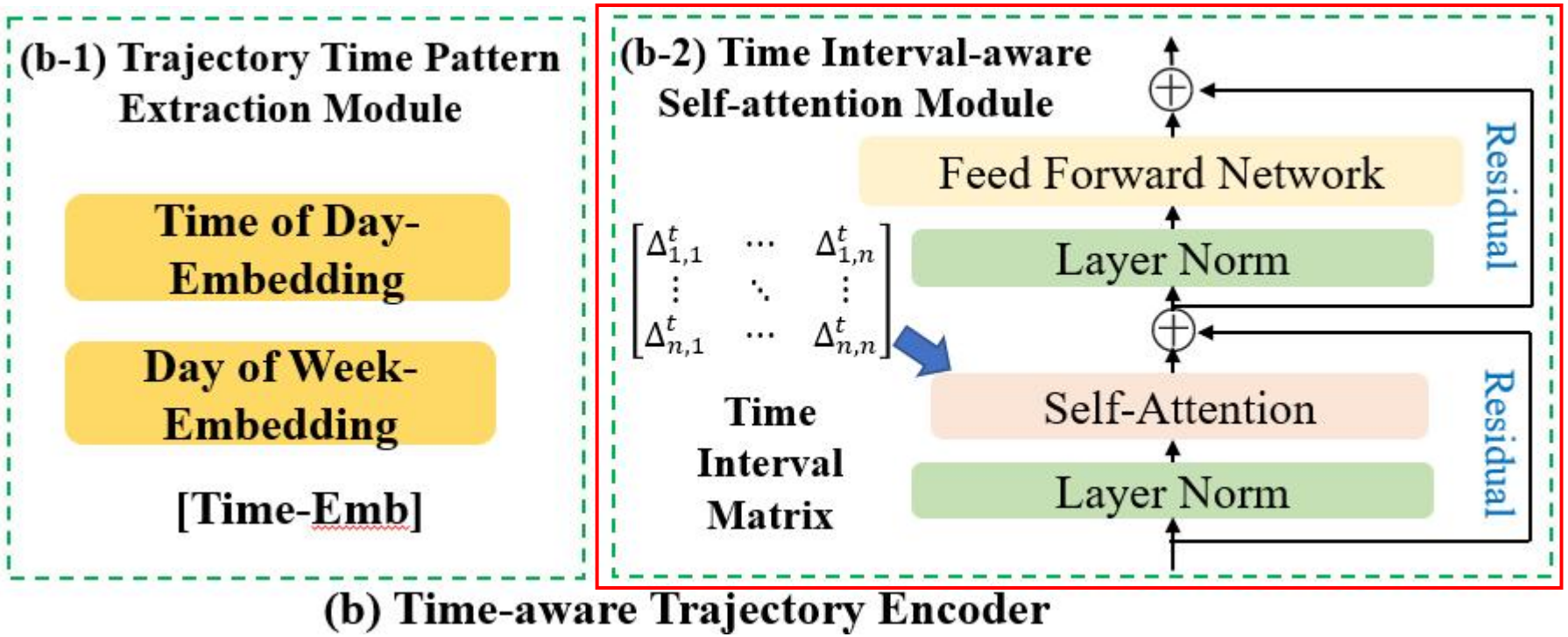
START: 时间感知的轨迹编码器（阶段 2）



- （b-1）轨迹时间模式抽取模块：

使用两个时间表征向量，分别编码time-of-day和day-of-week的时间周期性

START: 时间感知的轨迹编码器（阶段 2）

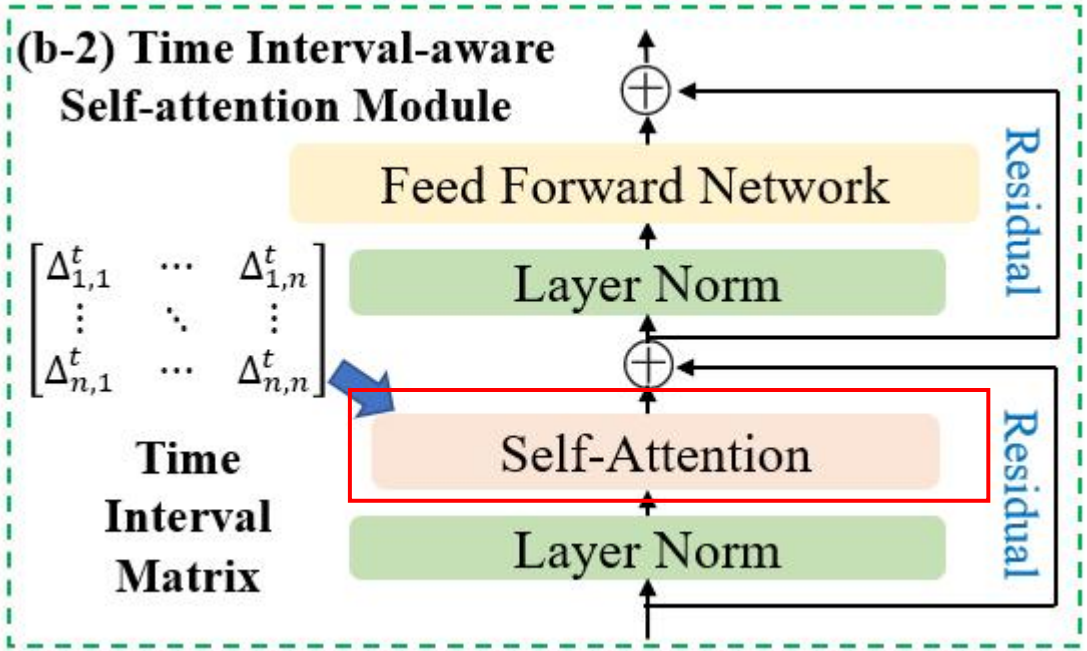


(b) Time-aware Trajectory Encoder

- （b-2）时间间隔感知的自注意力模块：

用于编码轨迹中的上下文信息，将轨迹序列中的所有路段的表征转换成轨迹的表征，并融合轨迹访问时间戳的不规则的间隔信息。

START: 时间感知的轨迹编码器（阶段 2）



时间间隔感知的自注意力模块

- 标准自注意力机制

$$A_h(\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) = \text{softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^T}{\sqrt{d'}}\right)\boldsymbol{V}_h.$$

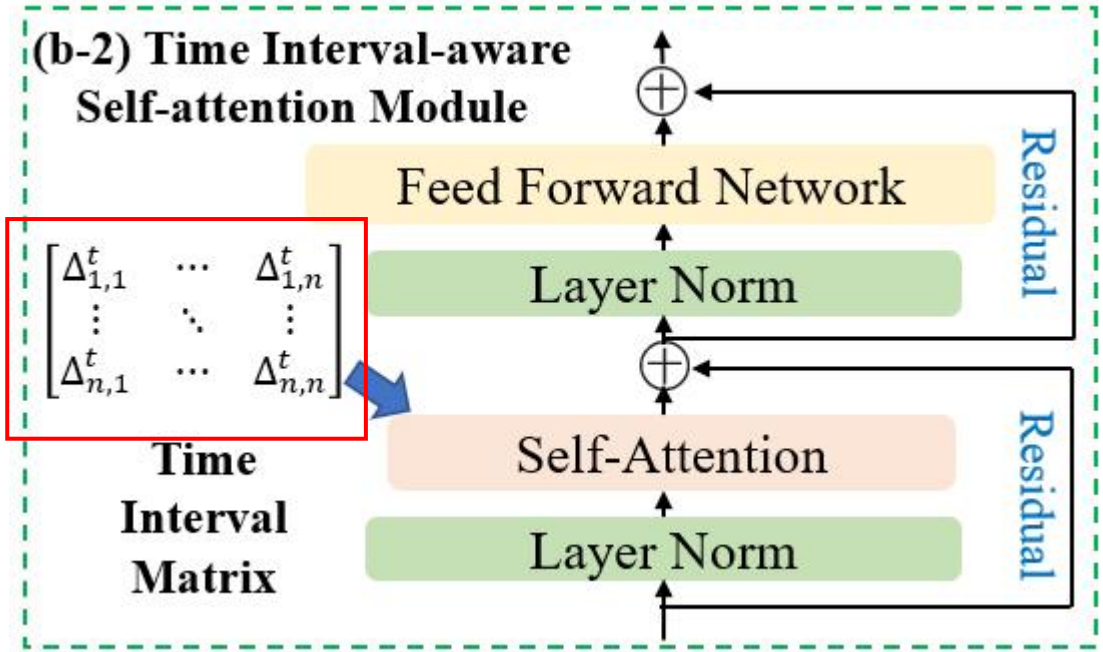
- 融合时间间隔信息的自注意力机制

$$TA_h(\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) = \text{softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^T}{\sqrt{d'}} + \tilde{\boldsymbol{\Delta}}\right)\boldsymbol{V}_h,$$

$\tilde{\Delta}$是自适应时间间隔矩阵，其中的每一个元素用于衡量轨迹中各路段之间的影响程度。

给定两个路段，其时间间隔越小则相互之间的影响就越大，也就是矩阵对应位置的值越大。

START: 时间感知的轨迹编码器（阶段 2）



时间间隔感知的自注意力模块

- 原始的时间间隔矩阵：

$$\delta_{i,j} = |t_i - t_j| \qquad \Delta = \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,|\mathcal{T}|} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,|\mathcal{T}|} \\ \cdots & \cdots & \cdots & \cdots \\ \delta_{|\mathcal{T}|,1} & \delta_{|\mathcal{T}|,2} & \cdots & \delta_{|\mathcal{T}|,|\mathcal{T}|} \end{bmatrix}.$$

- 引入减函数，实现“间隔越大，影响越小”

$$\delta'_{i,j} = 1/\log(\mathrm{e} + \delta_{i,j}),$$

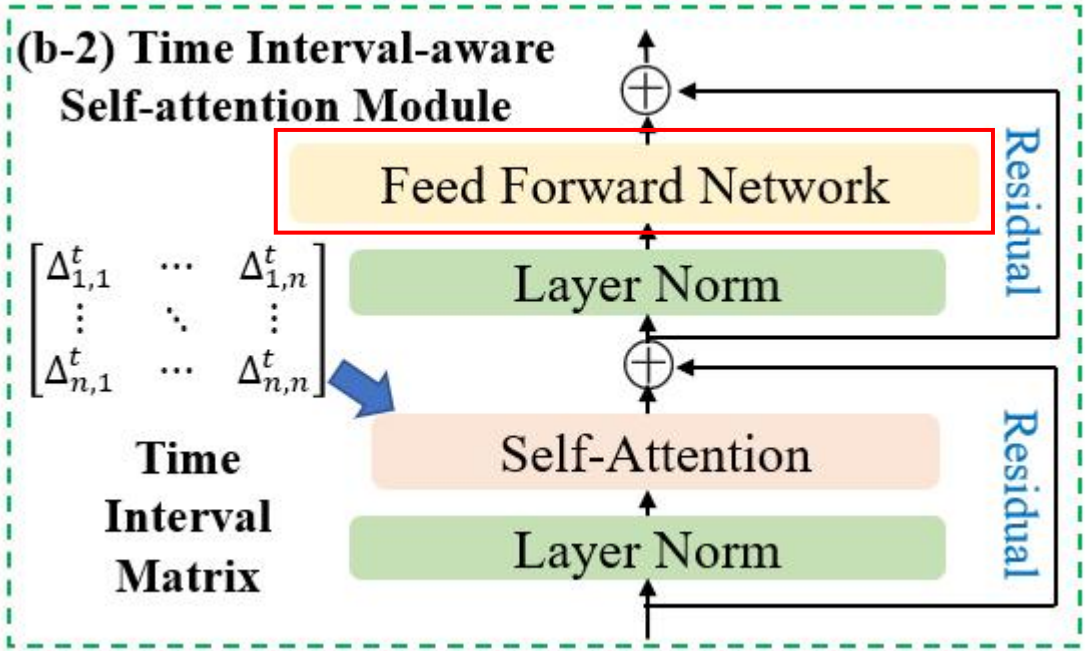
- 引入自适应参数，使得时间间隔矩阵可学习

$$\tilde{\delta}_{i,j} = (\mathrm{ReLU}(\delta'_{i,j}\,\boldsymbol{\omega}_1))\,\boldsymbol{\omega}_2^T,$$

- 把时间间隔信息注入到注意力的计算中

$$TA_h(\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^T}{\sqrt{d'}} + \tilde{\boldsymbol{\Delta}}\right)\boldsymbol{V}_h,$$

START: 时间感知的轨迹编码器（阶段 2）



时间间隔感知的自注意力模块

多头注意力：

连接多个头的输出，进行线性变化，得到注意力的输出

$$X' = \text{MultiAtt}(Q, K, V) = (TA_1 \| \ldots \| TA_{H_2})W^O.$$

前馈神经网络：

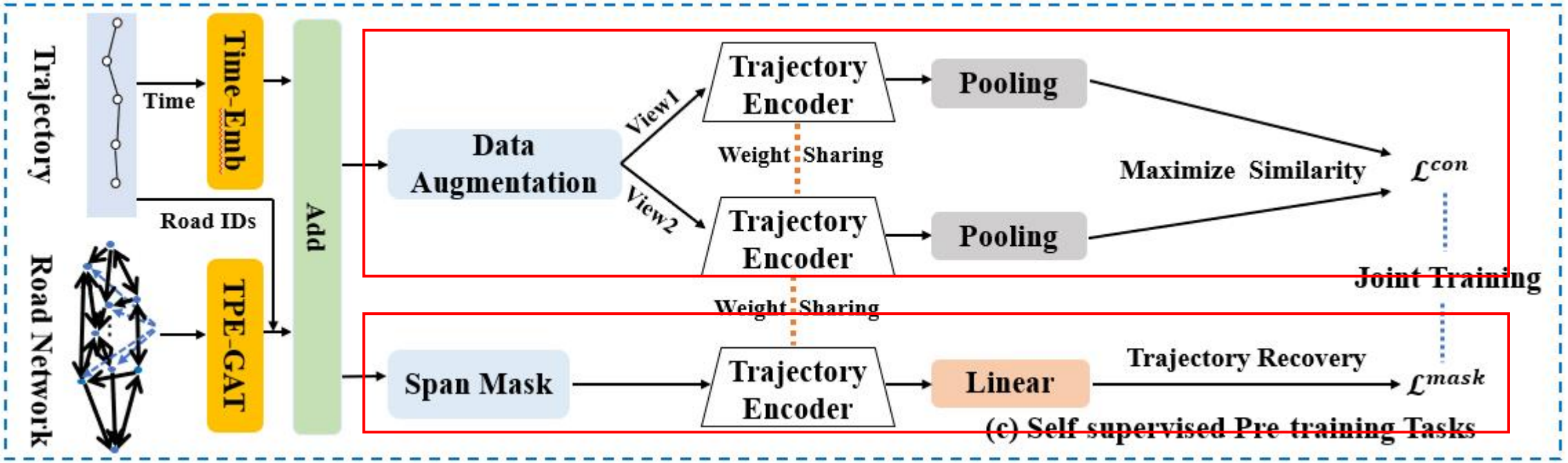
两层前馈神经网络结合$ReLU$激活函数得到最终输出

$$Z = (\text{ReLU}(X'W_F^1 + b_F^1))W_F^2 + b_F^2,$$

表征池化：

对顶层的路段表征序列$Z \in \mathbb{R}^{|T|*d}$进行平均池化，得到整条轨迹的表征向量 $p \in \mathbb{R}^d$。

START: 自监督训练任务



- 连续掩码预测：从轨迹中选取若干**连续的子序列**进行掩码，使用模型对掩码掉的路段进行预测。

- 轨迹对比学习：对比学习的主要目的是拉近语义相近的正样本之间距离，推远负样本的距离。通过考虑轨迹的时空特性，设计4种轨迹数据增强方案，生成对比学习的不同视图，通过对比学习，学习感知轨迹语义的表征。

START: 轨迹数据增强

- 轨迹截断：从轨迹的首或尾随机截断一部分的路段序列。（空间维度）
- 时间偏移：从轨迹中选取一部分路段，根据其历史平均通行时间，将对应的时间戳进行偏移。（时间维度）
- 轨迹掩码：从轨迹中选取若干连续的子序列进行时空的掩码。（时空维度）
- Dropout：通用的数据增强方案，对轨迹表征使用Dropout，即随机置为0。（时空维度）

START: Loss函数设计

- 连续掩码预测： 引入了一个全连接层来预测掩码的路段，使用交叉熵作为训练的loss。

$$\hat{\boldsymbol{Z}} = \boldsymbol{Z}\boldsymbol{W}_m + \boldsymbol{b}_m \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{V}|},$$

$$\mathcal{L}_{\mathcal{T}}^{mask} = -\frac{1}{|\mathcal{M}|}\sum_{v_i\in\mathcal{M}} \log \frac{\exp(\hat{\boldsymbol{Z}}_{v_i})}{\sum_{v_j\in\mathcal{V}}\exp(\hat{\boldsymbol{Z}}_{v_j})},$$

- 轨迹对比学习： 使用InfoNCE loss，根据Batch内的负样本计算对比学习的loss。

  Batch内的N个样本，通过数据增强得到N个增强样本，构成了N对正样本，而每一个样本，跟Batch内除了正样本之外的2N-2个样本构成负样本对。

$$\mathcal{L}_{i,j}^{con} = -\log \frac{\exp(\text{sim}(\boldsymbol{p}_i, \boldsymbol{p}_j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{[k\neq i]}\exp(\text{sim}(\boldsymbol{p}_i, \boldsymbol{p}_k)/\tau)},$$

- 联合预训练： $\mathcal{L}^{pre} = \lambda_1\mathcal{L}^{mask} + \lambda_2\mathcal{L}^{con},$

目录

CONTENTS

实验数据

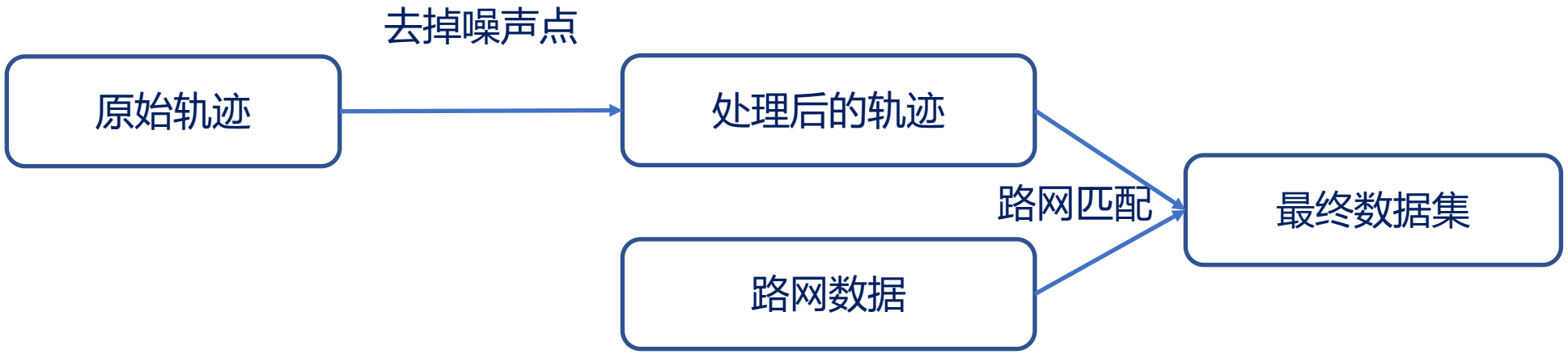


TABLE I
STATISTICS OF THE TWO DATASETS AFTER PREPROCESSING.

Dataset	BJ	Porto
Time span	2015/11/01-2015/11/30	2013/07/01-2014/07/01
#Trajectory	1018312	695085
#Usr	1677	435
#Road Segment	10903	38479
train/eval/test	656221/174478/187613	417040/139020/139025

下游任务

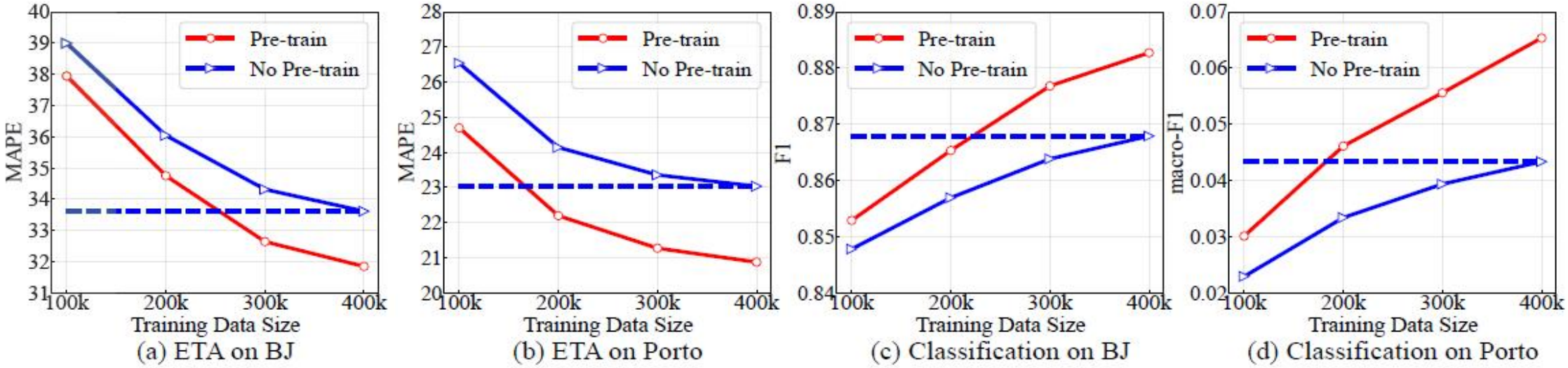
- **出行时间预测：** 实验设置：在轨迹表征的基础上，接入一个全连接层，来预测出行时间。
  微调：使用MSE进行微调。
  评估指标：MAE/RMSE/MAPE
- **轨迹分类：** 实验设置：根据特定的标签对轨迹进行分类，北京（载客状态二分类），波尔图（司机ID多分类）。在轨迹表征的基础上，接入一个全连接层，来预测类别。
  微调：使用交叉熵损失进行微调。
  评估指标：多分类（Micro-F1, Macro-F1, Recall@5）
  二分类（ACC, F1, AUC）
- **轨迹相似度计算：** 最相似轨迹查询和Top-k相似轨迹查询。
  微调：不需要微调。
  评估指标：最相似（Mean Rank, Hit Ratio@1, Hit Ratio@5）
  Top-k （Precision）即Top-k结果的覆盖率

性能实验

		Travel Time Estimation			Trajectory Classification			Most Similar Trajectory Search		
BJ	Models	MAE ↓	MAPE(%) ↓	RMSE ↓	ACC ↑	F1 ↑	AUC ↑	MR ↓	HR@1 ↑	HR@5 ↑
	traj2vec	10.13±0.12	37.95±0.92	56.83±0.47	0.811±6e-4	0.852±1e-3	0.873±2e-5	7.186±0.03	0.607±1e-4	0.766±7e-5
	t2vec	10.03±0.10	36.42±1.31	56.65±0.12	0.814±1e-3	0.863±1e-2	0.879±9e-4	5.948±0.01	0.788±3e-4	0.935±8e-5
	Trembr	9.997±0.11	34.20±0.88	36.97±0.38	0.818±1e-3	0.871±2e-3	0.880±2e-3	2.509±2e-3	0.884±5e-4	0.952±6e-5
	Transformer	10.74±0.48	39.61±1.52	57.16±0.56	0.794±2e-3	0.845±1e-3	0.846±1e-3	40.60±0.19	0.515±1e-4	0.649±1e-4
	BERT	10.21±0.14	37.31±0.17	37.09±0.36	0.804±3e-3	0.862±2e-3	0.864±3e-3	27.10±0.11	0.587±3e-3	0.712±4e-4
	PIM	10.19±0.09	39.04±0.58	57.73±0.26	0.803±1e-3	0.861±1e-3	0.862±9e-4	23.51±0.12	0.760±4e-3	0.898±2e-4
	PIM-TF	12.05±0.03	43.14±0.66	61.15±0.33	0.789±1e-3	0.849±2e-3	0.842±6e-3	86.45±0.32	0.296±2e-4	0.340±2e-4
	Toast	10.69±0.22	35.37±1.14	57.41±0.41	0.810±2e-3	0.870±2e-3	0.871±2e-3	29.53±0.15	0.611±2e-4	0.746±3e-4
	START	**9.134**±0.03	**30.92**±0.35	**35.40**±0.09	**0.853**±2e-3	**0.896**±1e-3	**0.916**±4e-4	**1.295**±1e-3	**0.969**±4e-4	**0.997**±4e-5
	Improve	8.63%	9.59%	4.24%	4.28%	2.87%	4.09%	48.39%	9.62%	4.73%
Porto	Models	MAE ↓	MAPE ↓	RMSE ↓	Micro-F1 ↑	Macro-F1 ↑	Recall@5 ↑	MR ↓	HR@1 ↑	HR@5 ↑
	traj2vec	1.552±6e-3	23.70±0.35	2.351±4e-3	0.063±3e-3	0.038±3e-3	0.183±5e-3	30.52±0.13	0.552±3e-4	0.732±5e-4
	t2vec	1.539±5e-3	23.65±0.12	2.324±5e-3	0.068±2e-4	0.048±3e-4	0.187±3e-4	12.70±0.08	0.746±3e-4	0.856±8e-4
	Trembr	1.480±2e-3	22.64±0.37	2.164±0.01	0.071±9e-4	0.049±1e-3	0.192±2e-3	4.635±1e-3	0.846±4e-4	0.929±8e-5
	Transformer	1.738±3e-3	25.72±0.26	2.637±2e-3	0.028±7e-3	0.018±5e-3	0.075±8e-3	68.58±0.21	0.447±2e-4	0.664±5e-5
	BERT	1.593±7e-3	24.63±0.57	2.291±3e-3	0.065±3e-4	0.044±1e-3	0.184±1e-3	39.12±0.15	0.511±4e-3	0.714±5e-4
	PIM	1.559±3e-3	24.68±0.25	2.339±0.01	0.061±4e-4	0.037±3e-4	0.153±5e-4	19.53±0.10	0.653±3e-4	0.774±7e-4
	PIM-TF	1.945±2e-3	28.82±0.15	2.841±3e-4	0.025±4e-3	0.016±5e-3	0.069±7e-3	78.78±0.24	0.384±2e-5	0.547±3e-5
	Toast	1.624±8e-3	24.63±0.33	2.445±5e-3	0.062±1e-3	0.035±4e-4	0.181±1e-3	22.61±0.12	0.684±2e-5	0.789±2e-5
	START	**1.334**±3e-3	**20.66**±0.14	**2.001**±1e-3	**0.089**±4e-4	**0.067**±2e-3	**0.244**±1e-3	**1.897**±1e-3	**0.921**±3e-4	**0.973**±6e-5
	Improve	9.86%	8.75%	7.53%	25.35%	36.73%	27.08%	59.07%	8.87%	4.74%

对比8个基线模型，有较大程度的提升！

预训练效果的验证（1）



通过对比**引入预训练后的模型**以及**不引入预训练而采用有监督训练的模型**发现：

预训练可以显著减少数据的使用

（1）使用同样的数据量，预训练模型始终优于有监督模型，数据越多现象越明显。

（2）预训练模型使用更少的数据量，即可以超过有监督模型（更多数据）的性能。

预训练效果的验证（2）

Models	Travel Time Estimation			Trajectory Classification		
	MAE	MAPE(%)	RMSE	Micro-F1	Macro-F1	Recall@2
No Pre-train Geolife	12.325	78.547	19.584	0.519	0.498	0.790
Pre-train Geolife	11.980	73.489	18.613	0.568	0.571	0.814
Porto-START	10.455	65.371	18.024	0.623	0.619	0.832
BJ-START	**9.995**	**64.331**	**17.183**	**0.669**	**0.665**	**0.887**
Porto-Trembr	15.200	80.294	23.223	0.507	0.468	0.728
BJ-Trembr	14.851	79.239	23.109	0.512	0.486	0.741

通过将在一个大规模数据集上预训练的模型，迁移到一个小数据上进行微调，发现：

（1）在北京和波尔图分别训练模型，迁移到Geolife数据集上，性能有显著提升

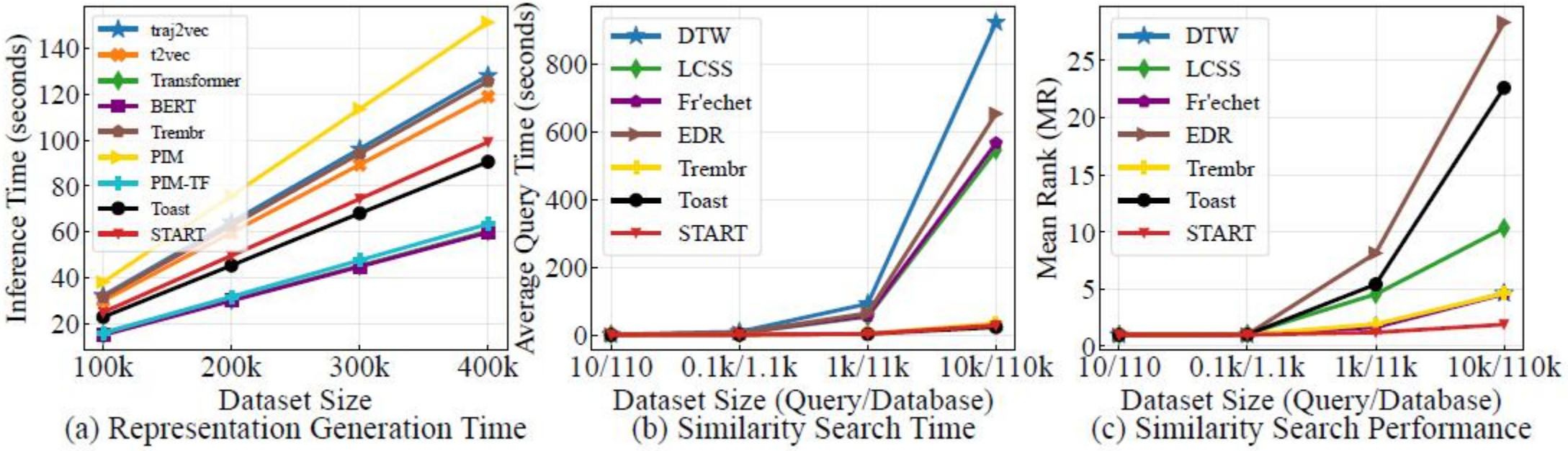
（2）在Geolife上训练模型，同样在Geolife上微调，性能有一部分提升

（3）基线模型Trembr的跨数据集迁移，性能无提升，甚至有副作用

预训练通过知识迁移，可以实现跨数据集甚至是跨路网结构的迁移，

以提升小数据集的性能，解决训练数据不足的问题

效率实验



- 轨迹编码效率： 仅需25s可以将10万条轨迹编码成向量进行存储。
- 相似度计算效率： 超过传统的相似度计算方法至少一个数量级。 (DTW/EDR/LCSS等)
- 相似度计算性能： 模型学到的轨迹表征不需要进行微调，就可以直接计算向量的欧氏距离以代表轨迹的相似度。相似度查询的准确率超过传统算法。 (DTW/EDR/LCSS等)

结论

- 我们提出了一个两阶段的轨迹表征学习方法，将时间规律和旅行语义融合到学习到的轨迹表征中。
- 我们设计了两个自监督预训练方法来训练我们的模型，相比于以往的方法，充分了考虑了轨迹的时空特性。
- 模型的性能有明显的优势，并且从实验上验证了所提出的预训练任务可以减少数据的使用，并且预训练后的模型可以实现跨数据集甚至跨路网结构的迁移，有助于解决现实应用的数据不足问题。

北京航空航天大学计算机学院
School of Computer Science and Engineering, Beihang University

谢谢观看！

姜佳伟
SY2106210