

# EPMC: An Enhanced Pre-training Model for Cellular Trajectory Representation Learning

Haihui Xu, Jiarong Zhou✉, Hai Hu, Jingwei Tu, Xiaoxiao Lv, and Chu Xu

China Mobile (Suzhou) Software Technology Co., Ltd, China  
{xuhaihui, zhoujiarong, huhai, tujingwei,  
lvxiaoxiao, xuchu}@cmss.chinamobile.com

**Abstract.** Trajectory representation learning (TRL) aims to map raw trajectories to  $d$ -dimensional vectors, which has been extensively studied on GPS-based trajectory and check-in sequences. The task of Cellular Trajectory Representation Learning (CTRL) is a new problem that focuses on analyzing trajectories derived from cellular-based data. To this end, we extend the task of TRL onto cellular trajectory by developing an enhanced pre-training model. Apart from the inherent sequential features involved in trajectories that have been widely exploited in existing work, the temporal characteristics and cellular-based network topological information are fully explored for better TRL in this paper. Specifically, the proposed method consists of two main components. Firstly, we build a time-aware encoding layer to incorporate the representations of travel preference and temporal periodic patterns in a cellular trajectory. Then, we build spatio-temporal dependencies encoding layer based on an enhanced pre-trained model to learn the representations of long-term dependencies, meanwhile incorporating spatial correlations with graph convolution networks. The experiments conducted on two real-world datasets demonstrate the effectiveness of our proposed model.

**Keywords:** Cellular Trajectory · Trajectory Representation Learning · Graph Convolution Network

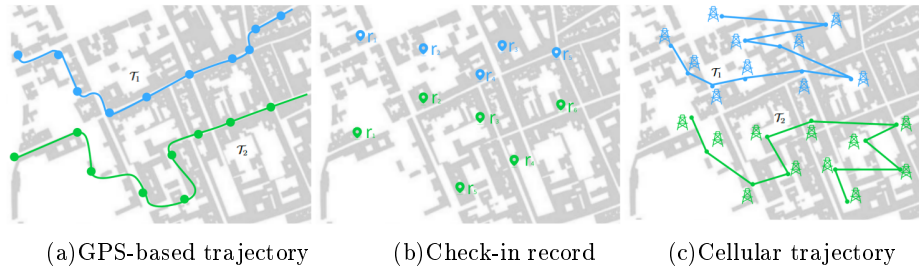
## 1 Introduction

With the widespread availability of smartphones and GPS-based devices, a substantial amount of spatio-temporal data, including cellular trajectories, GPS-based trajectories and check-in sequences, are generated [6]. This wealth of data facilitates a wide range of spatio-temporal data mining tasks [12].

Traditional studies on trajectory data analysis usually require manual feature engineering and task-specific models, making them challenging to adapt to different applications. To address the limitation, trajectory representation learning has emerged as a critical topic in recent years [14]. It aims to transform raw trajectory data into meaningful vectors that can be applied in various downstream tasks, rather than being confined to a single or specific task.

Some early studies utilize general seq2seq model to learning trajectory representation. These efforts consider spatio-temporal trajectories as ordinary se-

quence data and thus cannot fully capture spatial-temporal semantic information. After that, some methods like [2] introduce NLP technology to learn trajectory representation vectors. Subsequently, many methods are proposed for specific tasks, such as for trajectory similarity computation [15] and anomalous trajectory detection. However, most of existing trajectory representation learning methods focus on either GPS-based trajectories or check-in sequences, thus our study focuses on trajectory representation learning based on cellular trajectory in this paper. To illustrate the difference between above-mentioned three types of trajectory data, we present an example in Fig.1. GPS-based trajectories are



**Fig. 1.** An example of three types of trajectory data

typically closely linked to the road network. The check-in data formed by people uploading their real-time location information in location-based social networks face the challenge of sparse characteristic. Unlike GPS-based trajectories and check-in data, cellular trajectories which usually collected by communication companies, have more complete position sequences and contain dense information. Although, there has been some studies [8] [13] on cellular trajectories, they usually learn the trajectory representation for specific task, and cannot fully capture the critical spatial-temporal characteristics. To the best of our knowledge, there is no study which just learning a general representation vector of the cellular trajectory, not for specific tasks. Thus the resulting trajectory representation vectors can generalize and are convenient to migrate to other models or tasks. Additionally, it is crucial to capture various informational aspects of cellular trajectory data to learn representation vectors that are conducive to enhancing the performance of downstream tasks. Firstly, in the aspect of the temporal information, one of the key characteristics is the stay-time for travel preference. The stay-time data can reveal people’s preference and patterns of staying in the specific location, which is crucial for understanding people behavior and optimizing trajectory representation learning. Secondly, in terms of spatial information, one of the key characteristics is the spatial relationships within the topology of cell towers. The aforementioned existing methods ignore the fact that the location of a cell tower is a real geographic entity that can interact with devices and objects that connected to it.

To tackle the limitations discussed above, including the task-specific nature of TRL and the incomplete capture of spatio-temporal information of cellular trajectory, we focus on cellular trajectory representation learning to fill the gap

caused by the absence of the third type of trajectory data, and propose an **E**hanced **P**re-training **M**odel for **C**ellular **T**rajectory **R**epresentation **L**earning, namely **EPMC**. First, we develop a time-aware encoding layer to integrate the representations of stay-time and temporal periodic patterns within a cellular trajectory, a spatial context embedding layer to capture the position of cellular tower. Next, we construct a spatio-temporal pre-trained model based on the enhanced Transformer architecture with relative position encoding. This model learns the representations of time-dependent relationships, while also incorporating spatial correlations by utilizing a GCN-based encoder. In summary, the main contributions are as follows: (1) To the best of our knowledge, we are the first to explore the task of cellular trajectory representation learning. (2) We propose an enhanced pre-training model for TRL focus on cellular trajectory. By mainly designing a time-aware encoding layer and a GCN-based encoder to capture spatio-temporal information. (3) We report on an experimental study involving two real-world trajectory datasets for two downstream tasks. The experimental results prove the superiority and effectiveness of our model.

## 2 Related Work

Trajectory representation learning plays a crucial role in spatio-temporal data mining and analysis. Among the priori studies that introduce the concept of trajectory representation learning into data analysis is traj2vec [14]. Since then, many trajectory representation learning approaches have been proposed for specific downstream tasks. As a kind of sequential data, spatio-temporal trajectories share many common characteristics with sentences in natural language. Inspired by this idea, there are some existing methods based on words embedding model to lean only locations embedding. CTLE [13] employs MLM model to learn a contextual embedding for each location in a trajectory for the task of next location prediction. And there are some common methods that include sequence-based model, such as Trembr [3] and Toast [?].

In addition, some studies have designed self-supervised tasks to construct models. START [4] introduces contrastive learning with the MLM task with the goal of predicting the masked trajectory points to train the model to learn trajectory representations. These above existing studies mainly focus on learning embeddings for generally these types of spatio-temporal data, including location, road network, GPS-based trajectory and check-in sequence data.

On the other hand, only a few studies focus on cellular trajectory. GCMT [8] is a multitask framework to derive a meaningful intention representation for the task of both the spatio-temporal joint prediction. Theoretically, there is no study just to learn a general representation vectors for cellular trajectories, but not for specific task. To sum up, the above existing studies focus on cellular trajectory neglect that trajectory representation should be generalized and versatile. Hence, we introduce a novel framework for cellular trajectory representation by proposing an enhanced pre-training model to fully capture spatio-temporal information with in cellular trajectory.

### 3 Preliminaries

**Definition 1** *Cell Tower*. The cell tower is situated at a fixed geographical location, identified by its longitude and latitude coordinates. As the mobile phone moves, it connects to cell towers in sequence, generating cellular trajectory data.

**Definition 2** *Cellular Trajectory*. A cellular trajectory is a sequence of points recorded by cell tower denoted as  $\mathcal{T} = \langle x_1, x_2, \dots, x_n \rangle$ , where  $x_i = (u_i, pos_i, t_i, st_i)$  is a trajectory point under cellular-based positioning, which contains its user identifier  $u_i$ , position  $pos_i$ , timestamp  $t_i$  and stay-time  $st_i$ ,  $n$  is the length of the trajectory,  $pos_i = (lng_i, lat_i)$  denotes the the longitude and latitude of point. The  $pos_i$  is the location of the cell tower connected to the smartphone.

**Problem Formulation**. Given a set of cellular trajectory dataset  $T = \{\mathcal{T}_i\}_{i=1}^{|T|}$ , the Cellular Trajectory Representation Learning task aims to find a function or model, mapping each trajectory to a generic fixed-dimensional representation vector, denote as  $E_{\mathcal{T}} \in \mathcal{R}^d$ , where  $d$  denotes the vector dimension.

## 4 Proposed Model

### 4.1 Time-Aware Encoding

In this section, we capture temporal information in a cellular trajectory.

**RBFN-based Embedding**. Many existing approaches often utilize mean and standard deviation to normalize the training datasets. But these statistical features may be non-transferable when the test datasets shows significant heterogeneity of data and has no spatial overlap with the training data distribution. To address this challenge, we employ instance normalization(IN) to process the datasets. Instead of considering all data, this method calculate the mean and standard deviation of individual input instance  $X_i \in \mathcal{R}^T$  for each cellular trajectory, rather than relying on the global training data statistics. The formalization of this process is as follows:

$$X'_i = \text{IN}(X_i) = \frac{X_i - \mu}{\sigma_1} \quad (1)$$

where  $\mu$  and  $\sigma_1$  are respectively the mean and standard deviation of the input instance  $X_i$ . The study [5] indicates that instance normalization effectively mitigates distribution shifts between training and test sets.

Then we employ RBFN to construct temporal encoding layer for capturing stay-time information. The stay-time data can reveal users' habits and patterns of staying in specific locations, which is very important for understanding user behavior and optimizing trajectory representation learning. RBFN is a type of artificial neural network that utilize radial basis functions as activation functions. The radial basis functions commonly used in RBFN are typically Gaussian functions, defined as follows:

$$\phi(x) = \exp\left(-\frac{\|X'_i - C\|^2}{2\sigma_2^2}\right) \quad (2)$$

where  $X'_i$  is the input vector,  $C$  is the cluster center of the basis function, and  $\sigma_2$  is the standard deviation of the Gaussian function. Finally employ a linear transformation and RBFN encoding layer to get the stay-time for travel preference temporal information embedding  $E_{st} \in \mathcal{R}$ , and the formulation is as follows:

$$E_{st} = W_1 X' + \text{RBFN}(X') \quad (3)$$

where  $W_1$  is the learnable weight parameter.

**Temporal Context Embedding.** To capture the cellular trajectory patterns which influenced by multiple factors, our model effectively captures the distinct temporal patterns, such as periodic fluctuations caused by daily or weekly routines, as well as complex non-linear dependencies. Specifically, we extract features related to the timestamp of cellular trajectory points,  $t \in \mathcal{R}$ , the hour time of day,  $t_h \in \mathcal{R}$ , and the day of the week,  $t_w \in \mathcal{R}$ , and then generate time-related embeddings with linear layers in order to represent these temporal periodicity information. By explicitly modeling the nature features of cellular trajectory points, our approach can well learn trajectory periodic patterns and is well-equipped to improve the performance for downstream tasks. The final temporal periodicity information embeddings  $E_{tc}$  is as follows:

$$E_{tc} = W_2 t + \text{Concat}[W_3 t_h, W_4 t_w] \quad (4)$$

where  $W_2 \in \mathcal{R}^{T \times d}$ ,  $W_3, W_4 \in \mathcal{R}^{T \times d/2}$  are weight parameters and used to generate embeddings, which are combined with concatenation and addition.

## 4.2 Spatio-Temporal Dependencies Encoding

In this section, we then capture the spatial information in a cellular trajectory.

**Spatial Context Embedding.** To capture the location of a cell tower in cellular trajectory data, we incorporate the underlying spatial context cues within these cell towers. Firstly we employ min-max normalization method to normalize the longitude and latitude value into the range  $[0, 1]$ , and quantize the normalized longitude and latitude to integer index in the range  $[0, 1000)$ . Then obtain the corresponding embedding vectors through two embedding layers. Finally, we add these two embedding vectors to obtain the final spatial context information embedding  $E_{sc} \in \mathcal{R}^{T \times d}$ . The advantage of this method is that it can learn the complex relationships between longitude and latitude, and it can be input into the model along with other features for joint training.

**Enhanced Self-Attention.** To incorporate long-range dependencies of cellular trajectory, we build a pre-trained spatio-temporal model based on Transformer architecture to encode time-dependent relationship along with the spatial position information. The input embedding is as follows:

$$E_{in} = E_{st} + E_{tc} + E_{sc} \quad (5)$$

Then we construct the query vector  $Q^h = W_5^h(E_{in})$ , the key vector  $K^h = W_6^h(E_{in})$  and the value vector  $V^h = W_7^h(E_{in})$ , then calculate the attention score:

$$\text{Attention}(Q^h, K^h, V^h) = \text{softmax}\left(\frac{Q^h(K^h)^T}{\sqrt{d_h}}\right) V^h \quad (6)$$

where  $W_5^h, W_6^h, W_7^h$  are the weight parameters and  $d_h$  is the dimension.  $Q^h(K^h)^T$  is essentially a similarity matrix [10]. The position encoding forcibly binds each token to its absolute position. However, what truly affects sentence understanding is more often the relative positions between tokens.

To address above challenge, we use Rotary Position Encoding(RoPE) [9] to encode positional information into the input embeddings by using a rotation mechanism. Before computing the similarity matrix, the rotation is applied to both the query and key vectors, allowing the model to better effectively capture the relative positions between tokens, even when the sequence length changes.

$$\hat{Q}^h = \text{RoPE}(Q^h); \quad \hat{K}^h = \text{RoPE}(K^h) \quad (7)$$

After this, we can get the new query and key vectors which incorporated relative position information and the new attention score, the formulation is as follows:

$$\text{Attention}(Q^h, K^h, V^h) = \text{softmax} \left( \frac{\hat{Q}^h(\hat{K}^h)^T}{\sqrt{d_h}} \right) V^h \quad (8)$$

Thus the resulting temporal representation denotes as  $E_r$ , which can captures the periodic patterns and dynamic dependencies among different time segments.

**GCN-based Embedding.** It is important to capture spatial dependencies in a cellular trajectory data, as cell towers exhibit strong spatial correlations where the signal in one cell tower is often interfered with by the other neighboring ones. To incorporate the information, we firstly utilize geographic distance to construct the adjacency matrix of cell towers, and then employ graph convolution network to effectively capture spatial correlations between cell towers.

$$G_t = \delta [\alpha E_{r,t} + W_g M_{\text{adj}}(1 - \alpha) E_{r,t}] \quad (9)$$

where  $M_{\text{adj}}$  is the normalized adjacency matrix, and  $W_g$  is the weight parameter. The parameter  $\alpha$  balances original information preservation, and  $\delta$  is a dropout operation which helps mitigate over-fitting. Next, we use residual connections, RMSNorm and SwiGLU activation functions to obtain the output (i.e.,  $E_{o,t}^l$ ) of the  $i$ -th layer of spatio-temporal encoding network.

$$E_{o,t}^l = \text{SwiGLU}[\text{RMSNorm}(G_t^l + E_{o,t}^{l-1})] + G_t^l \quad (10)$$

The model captures spatio-temporal dependencies by stacking multiple layers, allowing it to learn the intricate relationships within cellular-based networks.

### 4.3 Fine-tuning for Downstream Tasks

We fine-tune our model use the flowing two downstream tasks.

**Trajectory Classification.** The task aims to classify trajectories based on a specific label. Here, we use user identifier as the label of a trajectory, and employ a fully connected layer with the softmax activation to obtain the predicted value as  $\hat{y} = \text{softmax}(\text{FC}(E_{\mathcal{T}}))$ . The cross-entropy loss is  $\mathcal{L}_{\text{CCE}} = \sum_{i=1}^C -y_i \log(\hat{y}_i)$ , where  $y$  is the ground truth and  $C$  is the number of categories.

**Travel Time Estimation.** This task aims to estimate the travel time from the origin to the destination with a given cell trajectory and the departure time. We build a regression model using a single fully connected layer to obtain the predicted value as  $\hat{y} = \text{FC}(E_{\mathcal{T}})$ . Then we use the mean square error as the optimization objective  $\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{j=1}^N (y_i - \hat{y}_i)^2$ , where  $y$  is the ground truth and  $N$  is the total number of trajectories in the test dataset.

## 5 Experiments

### 5.1 Datasets and Baselines

We use two real-world datasets derived from Guangzhou and Suzhou cities, where a cell tower that can provide signal to its surrounding area is regarded as a location. If a user enters the area covered by the cell tower, the switch time, stay-time and the longitude and latitude of location will be recorded. Notably, we remove the trajectory records with a stay-time below 10 minutes, and filter the trajectories with more than five staying records and users with more than 10 trajectories. To prove the superiority of our model, we compare it with the following models: **Word2vec** [7], **Traj2vec** [14], **TALE** [11], **Trembr** [3], **CTLE** [13], **Toast** [1] and **START** [4]. The statistics (i.e., #users, #trajectory points, #locations, #time span) of datasets (i.e., Guangzhou and Suzhou) are (9959, 1029022, 20865, 15 days) and (7358, 832196, 15120, 15 days) respectively.

### 5.2 Experiment Settings and Comparisons of Performance

For both two datasets, we split them into training, evaluation and test sets with an 6:2:2 ratio. We set the representation vectors dimension  $d$  as 256, the dropout ratio 0.1, the batch size 64 and  $\alpha$  is set to 0.05. We use the Adam optimizer with an initial learning rate of 0.001. For the task of the trajectory classification, we use Acc@k and macro-F1 as evaluation metrics. For the travel time estimation task, we use MAE, RMSE, and MAPE as evaluation metrics. We run five times of each method and take the average value as the final results.

**Comparisons of Performance.** The comparison results of our model with other baselines on two downstream tasks are shown in Table 1. The best result is shown in bold, we can make the following observations. (1) Our model EPMC performs better than baselines in terms of all metrics on these two tasks for the two real-world datasets. This demonstrates that EPMC’s pre-training process extracts rich spatio-temporal information from trajectories without additional task-specific supervision. (2) In detail, EPMC is superior to these methods (Word2vec, Traj2vec, Trembr, Toast, and START) for road segment or GPS-based trajectory, also better than these methods (TALE and CTLE) for location embedding or cellular-based trajectory, which is due to the fact that they ignore the important temporal stay-time for travel preference information and spatial cellular-based network topological information in a dense trajectory. (3) The

GPS-based representation learning method Traj2vec performs poorly, which is due to the fact that it ignores the noise and redundancy in the trajectory. START performs best among all baselines because it consider rich semantical information in trajectory. (4) We also notice the results in Suzhou are better than those in Guangzhou. The main reason is that Guangzhou is sparser than Suzhou, which has more users and locations than Suzhou. Such sparsity makes it hard for training a sequential model to learn mobility patterns.

**Table 1.** Performance comparison of methods on two datasets

	Task	Trajectory Classification			Travel Time Estimation		
	Metric Method	Acc@1 $\uparrow$	Acc@5 $\uparrow$	Macro-F1 $\uparrow$	MAE $\downarrow$	RMSE $\downarrow$	MAPE $\downarrow$
Guangzhou	Word2vec	0.3391	0.4846	0.1538	5.7611	6.6474	61.37%
	Traj2vec	0.3801	0.5262	0.1812	5.6597	6.5812	60.58%
	TALE	0.4158	0.5408	0.2329	5.2596	6.3047	59.55%
	Trembr	0.4587	0.5724	0.2791	4.7106	5.9841	57.13%
	CTLE	0.4962	0.6241	0.2868	4.6845	5.8713	55.24%
	Toast	0.4895	0.6206	0.2798	4.7017	5.9112	55.08%
	START	0.5573	0.6882	0.3076	4.5921	5.7841	48.18%
	EPMC	<b>0.6492</b>	<b>0.7801</b>	<b>0.3256</b>	<b>4.2416</b>	<b>5.6037</b>	<b>43.12%</b>
Suzhou	Word2vec	0.3861	0.5412	0.2259	5.0725	5.9676	59.32%
	Traj2vec	0.4428	0.5952	0.2663	4.5793	5.3861	51.05%
	TALE	0.4954	0.6308	0.3125	3.9127	5.0682	46.73%
	Trembr	0.5263	0.6575	0.3578	3.5217	4.5804	42.56%
	CTLE	0.5682	0.6986	0.3874	3.4156	4.4982	40.82%
	Toast	0.5702	0.6902	0.3778	3.4693	4.5721	41.27%
	START	0.6427	0.7648	0.3932	3.3046	4.4126	40.42%
	EPMC	<b>0.7117</b>	<b>0.8479</b>	<b>0.4328</b>	<b>3.1415</b>	<b>4.2062</b>	<b>38.78%</b>

### 5.3 Ablation Study

In the proposed model EPMC, we mainly design RBFN-based embedding layer to capture the stay-time for travel preference in a trajectory and introduce a graph convolution networks to incorporate spatial correlations between location of cell towers. The integration of them enhances our proposed model to extract more meaningful information from trajectories, ultimately improving the effectiveness of our model EPMC. In order to study the benefits brought by these two modules separately, we remove RBFN-based embedding layer from EPMC (the variant is denoted as w/o RBFN), and also remove GCN-based embedding layer (the variant is represented as w/o GCN).

Observed from Table 2, the model EPMC performs better than all variants, since in which both topological structure of cell towers and the travel preference characteristics of trajectories are taken into account. In contrast, only one of these two features is considered in w/o RBFN or w/o GCN. The results demonstrate the effectiveness and indispensability of RBFN-based embedding layer and GCN-based embedding layer for high-performance of model EPMC.

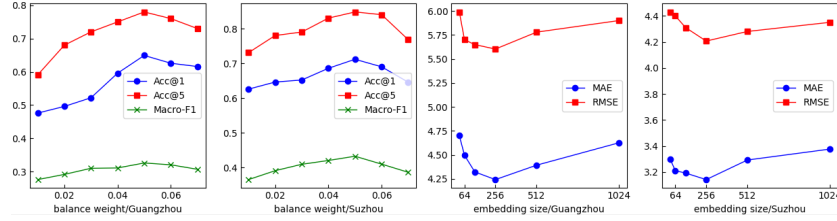


**Table 2.** Ablation experiment on two downstream tasks

	Task	Trajectory Classification			Travel Time Estimation		
	<div>Metric Method</div>	Acc@1↑	Acc@5↑	Macro-F1↑	MAE↓	RMSE↓	MAPE↓
Guangzhou	w/o RBFN	0.5024	0.6146	0.2708	4.7936	5.9561	55.86%
	w/o GCN	0.4952	0.6253	0.2601	4.7052	5.8632	56.74%
	EPMC	<b>0.6492</b>	<b>0.7801</b>	<b>0.3256</b>	<b>4.2416</b>	<b>5.6037</b>	<b>43.12%</b>
Suzhou	w/o RBFN	0.5861	0.7112	0.3842	3.4961	4.4056	40.06%
	w/o GCN	0.5562	0.6845	0.3718	3.5017	4.6216	41.24%
	EPMC	<b>0.7117</b>	<b>0.8479</b>	<b>0.4328</b>	<b>3.1415</b>	<b>4.2062</b>	<b>38.78%</b>

#### 5.4 Parameter Analysis

A crucial factor to be investigated is the balance weight parameter, as it balances original mainly temporal information preservation in attention with GCN-based embedding layer during pre-training process. We study the effect on trajectory classification task. Another is the dimension of trajectory representation which plays a crucial role in our model, we analysis it on travel time estimation task.

**Fig. 2.** Performance of EPMC vs. Parameter Analysis

The Fig.2 illustrates the relationship between the balance weights and the impact of embedding size. Observed from which, increasing the balance weight leads to improving model performance. This improvement can be attributed to the optimization of model parameters, which balances the preservation of primarily temporal information with the topological information of the spatial cellular networks, enabling the model to learn richer representations. However, increasing the balance weight beyond a certain point lead to disrupting the balance of capturing spatio-temporal information. It is evident that EPMC achieves the best performance when the embedding size is set to 256. With the increase of embedding size, more diverse features and patterns will be incorporated into representation vectors. However, we observe a decline in model performance when the embedding size exceeds 256, likely due to the introduction of excessive noise and redundant information.

## 6 Conclusion

In this paper, we propose an enhanced pre-training model for cellular trajectory representation learning. As existing methods focusing on embedding cellular tra-

jectory for the specific downstream task, neglecting to fully capture the temporal information and spatial relationships, we introduce stay-time as user’s travel preference in a cellular trajectory to improve performance of generalization ability, and build a RBFN-based embedding layer to capture it. Furthermore, we carefully design an enhanced pre-trained model to learn the representations of long-term dependencies, meanwhile capturing the position information of cellular towers, and incorporating spatial correlations with graph convolution networks in dense trajectories. Finally, extensive experimental results on two datasets demonstrate the effectiveness of our model.

## References

1. Chen, Y., Li, X., Cong, G., Bao, Z., Long, C., Liu, Y., Chandran, A.K., Ellison, R.: Robust road network representation learning: When traffic patterns meet traveling semantics. In: CIKM 2021. pp. 211–220
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
3. Fu, T.Y., Lee, W.C.: Trembr: Exploring road networks for trajectory representation learning. TIST **11**(1), 1–25 (2020)
4. Jiang, J., Pan, D., Ren, H., Jiang, X., Li, C., Wang, J.: Self-supervised trajectory representation learning with temporal regularities and travel semantics. In: 2023 IEEE 39th International Conference on Data Engineering. pp. 843–855 (2023)
5. Liu, X., Xia, Y., Liang, Y., Hu, J., Wang, Y., Bai, L., Huang, C., Liu, Z., Hooi, B., Zimmermann, R.: Largest: A benchmark dataset for large-scale traffic forecasting. ArXiv **abs/2306.08259** (2023)
6. Luca, M., Barlacchi, G., Lepri, B., Pappalardo, L.: Deep learning for human mobility: a survey on data and models (2020)
7. Rong, X.: word2vec parameter learning explained (2014)
8. Sang, Y., Xu, Y., Ning, B., Xie, Z.: Gcmt: a graph-contextualized multitask spatio-temporal joint prediction model for cellular trajectories. WWWJ **26**(4), 1649–1665 (2022)
9. Su, J., Ahmed, M., Liu, B.Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**(Feb.1), 127063.1–127063.12 (2024)
10. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv (2017)
11. Wan, H., Lin, Y., Guo, S., Lin, Y.: Pre-training time-aware location embeddings from spatial-temporal trajectories. TKDE **PP**(99), 1–1 (2021)
12. Wang, C., Lyu, F., Wu, S., Wang, Y., Xu, L., Zhang, F., Wang, S., Wang, Y., Du, Z.: A deep trajectory clustering method based on sequence-to-sequence autoencoder model. Transactions in GIS **26**, 1801 – 1820 (2022)
13. Yan Lin, Huaiyu Wan, S.G., Lin, Y.: Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In: AAAI. pp. 3210–3217 (2021)
14. Yao, D., Zhang, C., Zhu, Z., Huang, J., Bi, J.: Trajectory clustering via deep representation learning. IEEE (2017)
15. Zhou, S., Han, P., Yao, D., Chen, L., Zhang, X.: Spatial-temporal fusion graph framework for trajectory similarity computation. WWWJ **26**(4), 1501–1523 (2022)