Mutual Distillation Learning Network for Trajectory-User Linking

Wei Chen[1] , **Shuzhe Li**[1] , **Chao Huang**[2] , **Yanwei Yu**[1*] , **Yongguo Jiang**[1] , **Junyu Dong**[1]

[1]College of Computer Science and Technology, Ocean University of China
[2]Department of Computer Science, The University of Hong Kong

{weichen, lishuzhe}@stu.ouc.edu.cn, chaohuang75@gmail.com,
{yuyanwei, jiangyg, dongjunyu}@ouc.edu.cn

Abstract

Trajectory-User Linking (TUL), which links trajectories to users who generate them, has been a challenging problem due to the sparsity in check-in mobility data. Existing methods ignore the utilization of historical data or rich contextual features in check-in data, resulting in poor performance for TUL task. In this paper, we propose a novel Mutual distillation learning network to solve the TUL problem for sparse check-in mobility data, named *MainTUL*. Specifically, MainTUL is composed of a *Recurrent Neural Network (RNN) trajectory encoder* that models sequential patterns of input trajectory and a *temporal-aware Transformer trajectory encoder* that captures long-term time dependencies for the corresponding augmented historical trajectories. Then, the knowledge learned on historical trajectories is transferred between the two trajectory encoders to guide the learning of both encoders to achieve *mutual distillation of information*. Experimental results on two real-world check-in mobility datasets demonstrate the superiority of MainTUL against state-of-the-art baselines. The source code of our model is available at https://github.com/Onedean/MainTUL.

1 Introduction

The rapid development of Location-Based Social Network (LBSN) platforms has made it easier for humans to digitize their mobility behaviors by sharing their check-ins, opinions, and comments . These mobility behaviors can be used to understand and predict human movement patterns, facilitating intelligent business models and user experience. However, individual mobility is not always predictable due to the missing and sparsity of check-in data [Lian *et al.*, 2014]. Trajectory-user linking (TUL) [Gao *et al.*, 2017] is recently proposed as a task to identify user identities based on personal mobility trajectories. It plays an important role in revealing basic human movement patterns by mining user mobility behaviors. In addition, TUL could benefit a broad range of applications in business, transportation, epidemic prevention, and public safety, such as location-based services , tracking COVID-19 pandemic [Hao *et al.*, 2020], intelligent transportation [Dai *et al.*, 2021], and identifying terrorists/criminals for public safety [Huang *et al.*, 2018].

In this work, we are interested in linking trajectories to their potential users for check-in mobility data. TUL can essentially be seen as an extension of traditional trajectory classification tasks. Traditional trajectory measurement methods such as Longest Common Sub-Sequence (LCSS) and Dynamic Time Warping (DTW) can identify the most likely users by measuring the similarity between unknown trajectories and known trajectories. Recently, a handful of studies [Miao *et al.*, 2020; Zhou *et al.*, 2021a] have been developed for solving the TUL problem through deep trajectory representation learning. However, the existing methods still have three key limitations. *First,* all existing approaches still suffer from *data sparsity*, and perform poorly on sparse check-in mobility datasets. *Second,* existing methods only focus on spatial feature and/or temporal feature, and ignore the *rich contextual features* in check-in data such as POI categories. *Third,* most existing methods neglect the *utilization of historical data*. Due to the inherent sparsity of check-in data, the historical data of same users implies users‘ more complex movement patterns, which may help improve the model performance. Nevertheless, how to effectively utilize the knowledge from historical data remains a significant challenge.

To address the aforementioned challenges, we propose MainTUL, a Mutual distillation learning network model, to solve the TUL problem for sparse check-in trajectory data. In MainTUL, we design two different trajectory encoders – an RNN-based encoder to learn spatio-temporal movement patterns of input trajectory, and a temporal-aware transformer encoder to capture long-term time dependencies for the corresponding augmented trajectory. Then, the knowledge learned from the augmented trajectory data is transferred to guide the learning of RNN-based trajectory encoder. Meanwhile, input trajectory and augmented trajectory is exchanged to realize a mutual distillation learning network. Additionally, we also design a check-in embedding layer to produce multi-semantic check-in representations integrated with POI category and time information, which are then fed into the mutual distillation learning network. Experimental results on two real-life human mobility datasets show that our model significantly outperforms state-of-the-art baselines (**14.95**% Acc@1 gain

*Corresponding Author

and **14.11**% Macro-F1 gain on average) in TUL task.

Our contributions can be summarized as follows:

- We propose a mutual distillation learning network model, MainTUL, to solve TUL problem for sparse check-in trajectory data. Our MainTUL effectively leverages history data by trajectory augmentation and knowledge distillation to improve model performance.
- We design a temporal-aware transformer trajectory encoder in MainTUL to capture the long-term time dependencies in the augmented trajectories. With the designed trajectory encoders, our model achieves mutual distillation of information.
- We conduct extensive experiments on two real-life check-in mobility datasets. Results show that our model significantly outperforms state-of-the-art baselines by average 14.95% and 14.11% improvements in terms of Acc@1 and Macro-F1.

2 Related Work

Trajectory similarity measures have been widely used to explore the similarity of users from their spatiotemporal trajectories. Examples include LCSS [Ying *et al.*, 2010], DTW [Keogh and Pazzani, 2000], Spatio-Temporal Linear Combine distance [Shang *et al.*, 2017], and Spatiotemporal Signature [Jin *et al.*, 2019]. *However, such measures only consider spatial or spatio-temporal proximities and cannot capture the temporal dependencies in trajectory data.*

Recently, a variety of studies that focus on deep representation learning have been proposed for trajectory similarity computation [Li *et al.*, 2018; Yao *et al.*, 2020; Zhang *et al.*, 2020; Yang *et al.*, 2021]. *However, these studies focus more on improving the efficiency of trajectory similarity computation.* The introduction of TUL problem [Gao *et al.*, 2017; Zhou *et al.*, 2018] has further promoted the progress of deep trajectory representation learning in spatiotemporal data mining. Several methods [Miao *et al.*, 2020; Zhou *et al.*, 2021b] have been proposed to solve the TUL problem with deep neural networks. TULVAE [Zhou *et al.*, 2018] incorporates VAE model into TUL problem to learn hierarchical semantics of check-in trajectories in RNN. DeepTUL [Miao *et al.*, 2020] proposes recurrent networks with attention mechanism to model higher-order and multi-periodic mobility patterns by learning from historical data to alleviate the data sparsity problem. AdattTUL and TGAN [Zhou *et al.*, 2021b] introduce Generation Adversarial Network (GAN) to deal with the TUL problem. Recently, SML-TUL [Zhou *et al.*, 2021a] uses contrastive learning to learn the predictive representations from the user mobility itself constrained by the spatio-temporal factors. *Nevertheless, these methods use RNNs for modeling or prediction, which cannot effectively capture long-term time dependencies, and all methods ignore the effective use of historical data.*

[Hinton *et al.*, 2015] first proposed the concept of Knowledge Distillation (KD) in teacher-student architecture, which seeks to provide another pathway to gain knowledge about a task by training a model with a distillation loss in addition to the task loss. [Ruffy and Chahal, 2019] validate that appropriately tuned classical distillation in combination with a data augmentation training scheme provides orthogonal improvements. Recently, [Zhao *et al.*, 2021] propose a trainable mutual distillation learning model, which improves end-to-end performance more effectively than traditional teacher-student framework. *In this work, we are the first attempt to use mutual distillation strategy to effectively utilize knowledge extracted from historical data to improve TUL performance.*

3 Preliminaries

Definition 1 (Check-in Record)**.** *A check-in record is a triple $\langle u, t, p\rangle$ that represents user u visiting POI p at time t, where p denotes a uniquely identified venue in the form of $(id, category, \ell)$, ℓ representing the location of the POI.*

Definition 2 (Trajectory)**.** *A trajectory is a sequence of check-in records $(\langle u, t_1, p_1\rangle, \langle u, t_2, p_2\rangle, \ldots, \langle u, t_m, p_m\rangle)$ generated by user u in chronological order during a certain time interval τ, which is denoted by Tr^u_τ.*

A trajectory is called *unlinked*, if we do not know the user who generated it. The time interval τ in our work is set to 24 hours. We now state our problem as below:

Problem (Trajectory-User Linking)**.** *The task of TUL is to identify anonymous trajectories with the users who generate them. Let $\mathcal{T} = \{Tr_1, Tr_2, \ldots, Tr_n\}$ represent the set of unlinked trajectories, and $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ denote a set of users. Our goal is to find the mapping function $f(\cdot)$ satisfying the following condition:*

$$\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \|f(Tr_i) - y_i\|, f(Tr_i) \in \mathcal{U}, y_i \in \mathcal{U}, \quad (1)$$

where y_i is the true label of trajectory Tr_i, $\|\cdot\|$ represents difference evaluation operator (e.g., if $i = j$, $\|u_i - u_j\| = 0$, otherwise 1), and $\mathcal{F}$ is hypothesis space of TUL task.

4 Methodology

The architecture of MainTUL is presented in Figure 1. MainTUL contains three major components: *check-in embedding*, *trajectory encoder*, and *mutual distillation network*.

4.1 Multi-Semantic Check-in Embedding

This module contains two sub-modules: *trajectory augmentation* and *check-in embedding layer*.

Trajectory Augmentation

For check-in data, due to its inherent sparsity, the number of user check-ins in a sub-trajectory within a time window is very limited, while its long-term historical trajectory often implies more user movement patterns. Therefore, we explore two different trajectory augmentation strategies to generate the long-term trajectories for mutual distillation learning:

- *Neighbor Augmentation:* Given an input trajectory Tr in the time interval τ_i, we use $k+1$ sub-trajectories in time interval $[\tau_{i-k/2} : \tau_{i+k/2}]$ to form a long-term augmented trajectory in chronological order.
- *Random Augmentation:* For the input trajectory, we randomly sample k sub-trajectories from all trajectories of the corresponding user, and then form a long-term trajectory together with the input trajectory according to the time information.

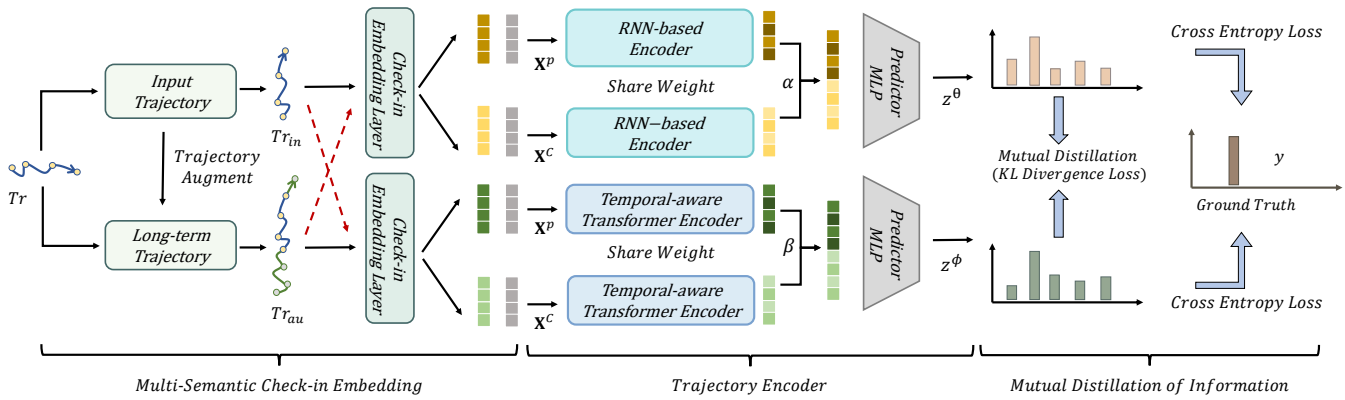

Figure 1: The architecture of the proposed MainTUL framework.

Check-in Embedding Layer

The goal of multi-semantic check-in embedding is to generate dense representations for each part in a check-in (such as location, category, and time slice) from sparse one-hot representations. In this way, we not only avoid the curse of dimensionality, but also capture contextual semantic information. For each trajectory, we actually have two sequence forms, one is a sequence of POIs, and the other is a sequence of POI categories. We divide a periodic time interval (*e.g.*, one day) into multiple time slices (*e.g.*, one hour for one time slice) and map check-in timestamps into corresponding time slices. Finally, we separately learn the embeddings for two sequences integrating with time slices as follows:

$$\begin{aligned} x_i^p &= tanh([\mathbf{W}_p p_i + b_p; \mathbf{W}_t t_i + b_t]), \\ x_i^c &= tanh([\mathbf{W}_c c_i + b_c; \mathbf{W}_t t_i + b_t]), \end{aligned} \tag{2}$$

where p_i, c_i and t_i are one-hot encodings for the i-th POI, its category and time slice in the sequences, and $\mathbf{W}_p$, $\mathbf{W}_c$, $\mathbf{W}_t$, b_p, b_c and b_t are learnable parameters.

4.2 Trajectory Encoder

To learn higher-order transition patterns of check-in trajectory sequences, trajectory encoders need to be designed. We note that the sequence lengths of input trajectories and augmented long-term trajectories differ greatly. Overly complex trajectory encoder is not suitable for processing shorter trajectories, and overly simple trajectory encoder cannot capture the long-term time dependencies of long trajectory sequences. Therefore, we design two different trajectory encoder based on RNN model and temporal-aware self-attention network respectively.

RNN-based Encoder

RNN is an efficient architecture for processing simple variable-length sequences. Due to the sparse nature of check-in trajectory, we use a popular RNN variant, Long Short-Term Memory network (LSTM) [Hochreiter and Schmidhuber, 1997; Huang *et al.*, 2019], as the encoder $f_\theta(\cdot)$ for processing the input trajectory. For both check-in POI and category sequences, we use a shared encoder $f_\theta(\cdot)$ and choose the hidden layer embedding of last time step as the representation of the two sequences. Then we use a learnable parameter α to balance the weights of the two representations and map them to user dimension via a multilayer perceptron (MLP) to obtain the final representation of the input trajectory:

$$z_{in}^\theta = MLP(\alpha \cdot f_\theta(\mathbf{X}_{in}^p) + (1-\alpha) \cdot f_\theta(\mathbf{X}_{in}^c)), \tag{3}$$

where $\mathbf{X}_{in}^p = \{x_i^p | i = 1, 2, \ldots, m\}$ and $\mathbf{X}_{in}^c = \{x_i^c | i = 1, 2, \ldots, m\}$ denote the embeddings of check-in POI sequence and category sequence of input trajectory, respectively.

Temporal-aware Transformer Encoder

The Transformer architecture [Vaswani *et al.*, 2017] is very expressive and flexible for both long- and short-term dependencies, which is proven to be superior to traditional sequence models in dealing with long sequences [Yang *et al.*, 2019; Wu *et al.*, 2020]. However, for a long trajectory sequence, it should be noted that there is still a problem, that is, the time slice information cannot fully reflect the information changes of the trajectory in the time dimension (*e.g.*, relative visit time difference). Therefore, inspired by [Zuo *et al.*, 2020; Lin *et al.*, 2020], we design a temporal-aware position encoder to replace the position encoder in the original transformer:

$$[PE(t_j)]_i = \begin{cases} \cos(w_i t_j), & \text{if } i \text{ is odd}, \\ \sin(w_i t_j), & \text{if } i \text{ is even}, \end{cases} \tag{4}$$

where w_i is a learnable parameter, i is the order of embedding dimension ($i \leq 2d$), and t_j is the visit timestamp for j-th check-in record.

Therefore, for any two POIs or categories in the sequence, the relative visit time information can be captured by:

$$PE(t_j)\, PE(t_j + \Delta_t)^\mathsf{T} = \sum_{i=1}^{d} \cos(w_i \Delta_t). \tag{5}$$

We use the improved temporal-aware transformer encoder as trajectory encoder $f_\phi(\cdot)$ for the augmented long-term trajectory. Similarly, a shared encoder $f_\phi(\cdot)$ is used to process the check-in POI and category sequences, and pooling the embedded tokens of last layer to obtain the latent representations of the two sequences. Next, we also use a learnable parameter β to balance the weights of the two representations

to obtain the final representation for the augmented long-term trajectory through a MLP predictor:

$$z_{au}^{\phi} = MLP(\beta \cdot f_{\phi}(\mathbf{X}_{au}^{p}) + (1-\beta) \cdot f_{\phi}(\mathbf{X}_{au}^{c})), \quad (6)$$

where $\mathbf{X}_{au}^{p}$ and $\mathbf{X}_{au}^{c}$ denote the embeddings of check-in POI sequence and category sequence of augmented long-term trajectory, respectively.

Notice that the detailed formulations of encoders f_θ and f_ϕ can be found at Appendix[1]

4.3 Mutual Distillation Network

Different from the traditional knowledge distillation [Hinton *et al.*, 2015], in this work, we do not strictly distinguish between teacher and student networks. Both are learning from scratch rather than compressing a new network from another deeper frozen model.

Let Tr_{in} and Tr_{au} represent the input trajectory and the augmented trajectory respectively. Trajectory encoder f_θ embeds Tr_{in} into z_{in}^{θ}. Similarly, Tr_{au} is encoded into z_{au}^{ϕ} via encoder f_ϕ. We expect encoder network f_θ can be trained similar to the true label y, and the knowledge of long-term trajectory with more representation ability can be transferred to f_θ. During the training, the loss function $\mathcal{L}_1$ is as:

$$\begin{aligned}\mathcal{L}_1 =& \mathcal{H}(y, z_{in}^{\theta}) + \mathcal{H}(y, z_{au}^{\phi}) + \\ & \lambda T^2 \,\mathrm{KL}(\varphi(z_{in}^{\theta}/T), \psi(z_{au}^{\phi}/T)),\end{aligned} \quad (7)$$

where $\mathcal{H}(\cdot,\cdot)$ refers to the conventional cross-entropy loss and $\mathrm{KL}(\cdot,\cdot)$ to the Kullback–Leibler divergence of softmax φ and log-softmax ψ. T is the temperature, intended to smooth outputs and λ is a balancing weight.

To maximize the use of training data, we propose a mutual distillation strategy, that is, exchange the input trajectory and the augmented trajectory for retraining, so that the two trajectory encoders can see more data to enhance the discriminative ability. In the mutual distillation strategy, encoders f_θ and f_ϕ are trained collaboratively and treated as peers rather than student/teacher. Specifically, for input trajectory Tr_{in} and augmented trajectory Tr_{au}, we swap and send them to different encoders to obtain new representations z_{in}^{ϕ} and z_{au}^{θ}, and calculate the loss function $\mathcal{L}_2$ as follows:

$$\begin{aligned}\mathcal{L}_2 =& \mathcal{H}(y, z_{in}^{\phi}) + \mathcal{H}(y, z_{au}^{\theta}) + \\ & \lambda T^2 \,\mathrm{KL}(\varphi(z_{in}^{\phi}/T), \psi(z_{au}^{\theta}/T)).\end{aligned} \quad (8)$$

This operation can also be regarded as a data augmentation strategy. Although the RNN encoder is not specially designed for long-term trajectory data, it is also beneficial to use more trajectory sequences related to the input trajectory during training. Our final loss function for optimizing mutual distillation network is:

$$\mathcal{L}_{total} = \mathcal{L}_1 + \mathcal{L}_2. \quad (9)$$

Notice that in TUL prediction stage, there is no data augmentation operation. That is, in testing, each input trajectory is encoded by trajectory encoder f_θ, and then is linked with the predicted user label.

Datasets	#users	#trajectories	#POIs	#categories	Duration
Foursquare	800	104,413	39,698	239	11 months
	400	51,969	24,526	231	11 months
Weeplaces	800	152,583	24,649	1,373	7 years
	400	75,873	18,482	1,177	6 years

Table 1: Statistics of the datasets.

5 Experiments

5.1 Datasets

We use two real-world check-in mobility datasets [Liu *et al.*, 2014; Yang *et al.*, 2015] collected from two popular location-based social network platforms, *i.e.*, Foursquare[2] and Weeplaces[3]. For Foursquare and Weeplaces, we choose top 800 and 400 users with the most check-ins for evaluating model performance respectively. In experiments, we use the first 80% of sub-trajectories of each user for training and the remaining 20% for testing, and select 20% training data as the validation set to cooperate with the early stop mechanism to find the best parameters and avoid overfitting.

The statistics of two datasets are summarized in Table 1.

5.2 Baselines

We consider the following baselines for comparison.

- **Classical methods:** (1) **LCSS** – a common and effective trajectory similarity measure method [Ying *et al.*, 2010]. (2) Signature Representation (**SR**) – a state-of-the-art trajectory similarity measure for moving object linking [Jin *et al.*, 2019; Jin *et al.*, 2020].
- **Machine learning methods:** (3) Linear Discriminant Analysis (**LDA**) – a classic spatial data classification method [Shahdoosti and Mirzapour, 2017]. (4) Decision Tree (**DT**) – an effective classification method for trajectory data [Jiang, 2018].
- **Deep neural network models:** (5) **TULER** – an RNN model for TUL task [Gao *et al.*, 2017], including three variants: RNN with Gated Recurrent Unit (**TULER-G**), LSTM (**TULER-L**) and bidirectional LSTM (**Bi-TULER**). (6) **TULVAE** – It utilizes VAE to learn the hierarchical semantics of trajectory in RNN [Zhou *et al.*, 2018]. (7) **DeepTUL** – a recurrent network with attention mechanism for TUL task [Miao *et al.*, 2020].

5.3 Evaluation Metrics and Parameter Settings

We use the Acc@k, Macro-Precision, Macro-Recall and Macro-F1 to evaluate the model performance. Specifically, Acc@k is used to evaluate the accuracy of TUL prediction.

For baselines, we use the parameter settings recommended in their papers and fine-tune them to be optimal. For MainTUL, we set check-in embedding dimension d to 512, λ to 10, use early stopping mechanism, and set patience to 3 to avoid over fitting. The learning rate is initially set to 0.001

[1]https://github.com/Onedean/MainTUL/tree/main/appendix

[2]http://sites.google.com/site/yangdingqi/home/foursquare-dataset

[3]http://www.yongliu.org/datasets.html

Dataset	Methods	Acc@1	Acc@5	Macro-P	Macro-R	Macro-F1	Acc@1	Acc@5	Macro-P	Macro-R	Macro-F1
		$\|\mathcal{U}\| = 400$					$\|\mathcal{U}\| = 800$				
Foursquare	TULER-L	52.88%	64.08%	58.96%	50.69%	52.75%	47.36%	59.11%	52.52%	45.28%	46.73%
	TULER-G	52.41%	63.41%	58.39%	50.17%	52.09%	47.67%	58.94%	52.39%	45.53%	46.55%
	Bi-TULER	51.59%	63.02%	58.46%	49.45%	51.62%	49.03%	60.48%	54.92%	47.10%	48.51%
	TULVAE	52.61%	63.33%	58.75%	50.51%	52.66%	47.71%	59.55%	53.20%	45.65%	47.09%
	DeepTUL	53.99%	65.22%	61.78%	52.00%	54.65%	45.41%	58.20%	57.75%	43.33%	45.43%
	MainTUL	**60.58%**	**70.22%**	**62.91%**	**58.49%**	**59.12%**	**56.92%**	**67.85%**	**59.20%**	**55.02%**	**55.37%**
Weeplaces	TULER-L	38.44%	49.60%	41.54%	37.50%	37.33%	36.12%	47.56%	40.62%	35.27%	35.78%
	TULER-G	38.84%	50.17%	42.56%	37.76%	38.09%	36.35%	47.78%	40.63%	35.40%	35.81%
	Bi-TULER	38.15%	50.21%	43.23%	37.19%	37.77%	36.23%	47.93%	40.89%	35.29%	35.60%
	TULVAE	38.98%	50.60%	42.59%	38.11%	38.36%	35.23%	46.84%	38.87%	34.32%	34.51%
	DeepTUL	33.48%	46.34%	46.50%	33.08%	34.37%	26.65%	37.73%	37.22%	25.66%	26.24%
	MainTUL	**45.31%**	**58.28%**	**49.81%**	**44.24%**	**45.22%**	**41.90%**	**55.51%**	**46.63%**	**40.58%**	**41.62%**

Table 2: Performance comparison with deep neural network models. Macro-P/R: Macro-Precision/Recall

and decays by 10% every 5 epochs. We repeat 10 runs for each experiment and report the average for all methods. More experimental settings can be found in the appendix.

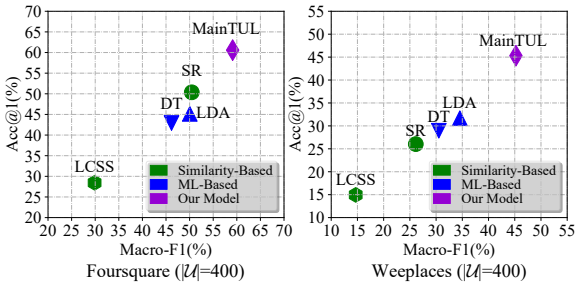

Figure 2: Performance comparison with classical models.

5.4 Overall Performance

We report the overall performance with deep neural network models in Table 2, where the best is shown in bold and the second best is shown as underlined. The comparison with classical learning methods are shown in Figure 2.

As shown in Table 2 and Figure 2, MainTUL significantly outperforms all baselines in terms of all evaluation metrics on two real-world check-in datasets. Specifically, MainTUL achieves average 14.95% Acc@1 and 14.11% Macro-F1 improvement in comparison to the bested performed baseline on two datasets. The main reason is that our designed mutual distillation model based on two different trajectory encoders captures the spatio-temporal movement patterns of users' check-in trajectories more effectively than RNN-based models (*e.g.*, TULER and DeepTUL). In addition, contextual features such as POI category and temporal information are integrated in MainTUL to further improve the performance.

We also observe that model performance on data with more users is worse than that on data with fewer users. This is intuitive because the more users the more difficult the classification becomes. However, the performance of our MainTUL is only reduced by 3.68% in Macro-F1 from $|\mathcal{U}| = 400$ to $|\mathcal{U}| = 800$, while state-of-the-art model (*i.e.*, DeepTUL) is reduced by 8.13%-9.22%. For DeepTUL, considering the historical data of all users does have a certain improvement on the data with fewer users, but when the number of users is large, a large amount of history data will also bring more noise, resulting in a sharp drop in performance. However, our model only uses the historical data of same user for trajectory enhancement and knowledge distillation during training, and does not require historical data for testing, and thus still performs better on the data with more users.

Notice that SML-TUL [Zhou *et al.*, 2021a] and TGAN [Zhou *et al.*, 2021b] are not compared in our experiments due to no publicly available source codes. However, our MainTUL significantly outperforms SML-TUL and TGAN in terms of Acc@k and Macro-F1 on Foursquare according to the results reported in [Zhou *et al.*, 2021a], even if MainTUL links more users (*e.g.*, MainTUL vs. SML-TUL vs. TGAN: 60.58% vs. 57.23% vs. 53.00% in Acc@1, and 59.12% vs 52.66% vs. 47.76% in Macro-F1 on Foursquare).

5.5 Ablation Study

Component Ablation

In our ablation study, we compare our model with the following three carefully designed variations: (1) **TUL-CA** – This variation removes the category and time information in check-in embedding layer. (2) **TUL-TA** – It uses position encoding in [Vaswani *et al.*, 2017] to replace our proposed temporal-aware position encoding. (3) **TUL-MUT** – It removes loss function $\mathcal{L}_2$ to prove the importance of extracting knowledge from each other.

The results of ablation study are shown in Figure 3. As we can see, the key components all contribute to performance improvement of our model. We can observe that TUL-MUT performs the worst in most cases, indicating that the mutual distillation strategy has the greatest impact on the performance improvement of our model. The comparisons between TUL-CA, TUL-TA and MainTUL reflects the importance of the contextual features and temporal-aware position encoding, respectively. The results in Figure 3 demonstrate that temporal-aware position encoding has a greater impact on model performance on Foursquare dataset, while contextual features have a greater effect on Weeplaces dataset.

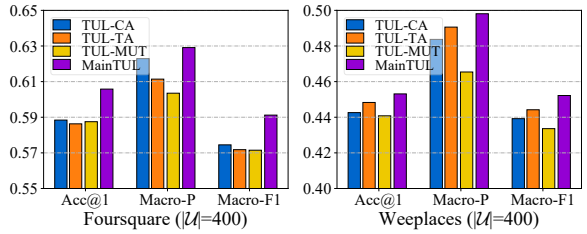

Figure 3: Experimental results of component ablation.

Loss function	Foursqaure Acc@1/Mac-F1	Weeplaces Acc@1/Mac-F1
original	60.58%/59.12%	45.31%/45.22%
w/o all λT^2 KL$(\cdot)$	55.36%/53.57%	40.01%/39.24%
w/o $\mathcal{H}(y, z_{in}^{\theta})$ and $\mathcal{H}(y, z_{in}^{\phi})$	60.06%/58.64%	43.49%/43.58%

Table 3: Loss function explorations. Mac-F1: Macro-F1.

Loss Function Ablation

We also evaluate the effectiveness of each term in our loss function (Eq. (9)). Table 3 depicts the experimental results of different loss functions on two datasets with 400 users.

First, we can see that removing term λT^2 KL$(\cdot)$ leads to the decrease in performance, which demonstrates the importance of dark knowledge of long-term augmented trajectory. Second, we notice that removing term $\mathcal{H}(y, z_{in}^{\theta})$ and $\mathcal{H}(y, z_{in}^{\phi})$, i.e., without considering the input trajectory labels, the model performance does not drop significantly. This indicates that our model can still achieve better results only through the knowledge distillation of augmented trajectories.

5.6 Parameter Study

We also evaluate the sensitivity of our MainTUL with respect to different settings of temperature T and hyperparameter λ in our loss function. The results on Foursquare are shown in Figure 4. As we can see, the performance first increases and then decreases, as T increases. This is intuitive because lower temperature the distribution more sharp, and higher temperature makes the distribution unable to extract effective information. The same observation is also presented on hyperparameter λ. In addition, it is clear that the performance increases rapidly with λ increasing from 0.1. This suggests our proposed knowledge distillation module contributes a lot to the overall performance.

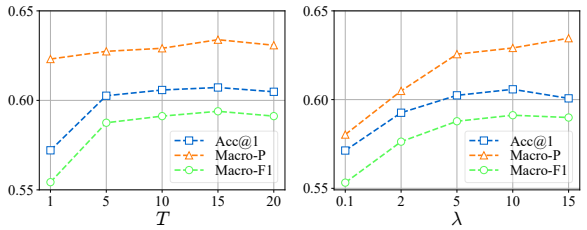

Figure 4: Parameter sensitivity *w.r.t.* T and λ on Foursquare.

5.7 Strategy Study

Effect of Trajectory Encoder Selection

We also evaluated the combination selection of different types of trajectory encoders. The results on Foursquare are shown in Table 4. We can conclude that when using the same type of trajectory encoders at the same time, the model performance is poor, which validates the rationality of our design of two different types of trajectory encoders. In the case of using two different encoders (*i.e.*, f_θ uses RNN encoder and f_ϕ adopts transformer encoder), using encoder f_θ for testing can achieve the best performance, which also shows that the simple encoder is more suitable for capturing the movement patterns of sparse sub-trajectories.

Train				Test		Acc@1	Mac-F1
f_θ		f_ϕ		f_θ	f_ϕ		
R-based	T-based	R-based	T-based				
✓		✓		✓		56.47%	55.03%
✓		✓			✓	56.55%	55.23%
✓			✓	✓		60.58%	59.12%
✓			✓		✓	57.87%	55.83%
	✓	✓		✓		57.60%	55.75%
	✓	✓			✓	59.39%	58.22%
	✓		✓	✓		53.80%	53.56%
	✓		✓		✓	53.80%	53.94%

Table 4: Effect of trajectory encoder selection. R-based: RNN-based, T-based: Transformer-based.

Effect of Augmentation Strategy

Finally, we evaluate the proposed two trajectory augmentation strategies. Based on the results on Foursquare in Table 5, we have two observations: (1) Both data augmentation strategies help improve model performance. (2) The random augmentation is better than the neighbor augmentation. The reason is that the random strategy may obtain the potential movement patterns of users by combining users' past travels randomly, resulting in better results.

k	0	2	4	8	16
Neighbor	51.09%	53.78%	54.01%	53.49%	53.51%
Random	51.09%	54.25%	59.14%	59.12%	59.00%

Table 5: Effect of augmentation strategy in Macro-F1.

6 Conclusion

In this paper, we propose a novel mutual distillation learning network (MainTUL) to solve TUL problem for sparse check-in mobility data. MainTUL effectively learns user movement patterns for input trajectory by the designed mutual distillation network consisting of two different trajectory encoders with multi-semantic check-in embeddings. Experiments on two real-world check-in mobility datasets demonstrate that MainTUL significantly outperforms state-of-the-art baselines in terms of all evaluation metrics for TUL prediction.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under grant Nos. 62176243, 61773331 and 41927805, the National Program on Key Research Project under grant No. 2019YFC1509100, and the National Key Research and Development Program of China under grant No. 2018AAA0100602.

References

[Dai *et al.*, 2021] Shaojie Dai, Jinshuai Wang, Chao Huang, et al. Temporal multi-view graph convolutional networks for citywide traffic volume inference. In *ICDM*, pages 1042–1047. IEEE, 2021.

[Gao *et al.*, 2017] Qiang Gao, Fan Zhou, Kunpeng Zhang, et al. Identifying human mobility via trajectory embeddings. In *IJCAI*, volume 17, pages 1689–1695, 2017.

[Hao *et al.*, 2020] Qianyue Hao, Lin Chen, Fengli Xu, and Yong Li. Understanding the urban pandemic spreading of covid-19 with real world mobility data. In *KDD*, pages 3485–3492, 2020.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Huang *et al.*, 2018] Chao Huang, Junbo Zhang, et al. Deepcrime: Attentive hierarchical recurrent networks for crime prediction. In *CIKM*, pages 1423–1432, 2018.

[Huang *et al.*, 2019] Chao Huang, Chuxu Zhang, Jiashu Zhao, et al. Mist: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting. In *WWW*, 2019.

[Jiang, 2018] Zhe Jiang. A survey on spatial prediction methods. *TKDE*, 31(9):1645–1664, 2018.

[Jin *et al.*, 2019] Fengmei Jin, Wen Hua, et al. Moving object linking based on historical trace. In *ICDE*, pages 1058–1069. IEEE, 2019.

[Jin *et al.*, 2020] Fengmei Jin, Wen Hua, et al. Trajectory-based spatiotemporal entity linking. *TKDE*, 2020.

[Keogh and Pazzani, 2000] Eamonn J Keogh and Michael J Pazzani. Scaling up dynamic time warping for datamining applications. In *KDD*, pages 285–289, 2000.

[Li *et al.*, 2018] Xiucheng Li, Kaiqi Zhao, Gao Cong, et al. Deep representation learning for trajectory similarity computation. In *ICDE*, pages 617–628. IEEE, 2018.

[Lian *et al.*, 2014] Defu Lian, Yin Zhu, Xing Xie, et al. Analyzing location predictability on location-based social networks. In *PAKDD*, pages 102–113. Springer, 2014.

[Lin *et al.*, 2020] Yan Lin, Huaiyu Wan, et al. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *AAAI*, 2020.

[Liu *et al.*, 2014] Yong Liu, Wei Wei, Aixin Sun, et al. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM*, pages 739–748, 2014.

[Miao *et al.*, 2020] Congcong Miao, Jilong Wang, Heng Yu, et al. Trajectory-user linking with attentive recurrent network. In *AAMAS*, pages 878–886, 2020.

[Ruffy and Chahal, 2019] Fabian Ruffy and Karanbir Chahal. The state of knowledge distillation for classification. *arXiv preprint arXiv:1912.10850*, 2019.

[Shahdoosti and Mirzapour, 2017] Hamid Reza Shahdoosti and Fardin Mirzapour. Spectral–spatial feature extraction using orthogonal linear discriminant analysis for classification of hyperspectral data. *European Journal of Remote Sensing*, 50(1):111–124, 2017.

[Shang *et al.*, 2017] Shuo Shang, Lisi Chen, Zhewei Wei, et al. Trajectory similarity join in spatial networks. *VLDB*, 10(11), 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[Wu *et al.*, 2020] Xian Wu, Chao Huang, Chuxu Zhang, et al. Hierarchically structured transformer networks for fine-grained spatial event forecasting. In *WWW*, pages 2320–2330, 2020.

[Yang *et al.*, 2015] Dingqi Yang, Daqing Zhang, et al. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *Transactions On SMC*, 45(1):129–142, 2015.

[Yang *et al.*, 2019] Zhilin Yang, Zihang Dai, Yiming Yang, et al. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*, 32, 2019.

[Yang *et al.*, 2021] Peilun Yang, Hanchen Wang, Ying Zhang, et al. T3s: Effective representation learning for trajectory similarity computation. In *ICDE*, pages 2183–2188. IEEE, 2021.

[Yao *et al.*, 2020] Di Yao, Gao Cong, Chao Zhang, et al. A linear time approach to computing time series similarity based on deep metric learning. *TKDE*, 2020.

[Ying *et al.*, 2010] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, et al. Mining user similarity from semantic trajectories. In *SIGSPATIAL Workshop on LBSNs*, pages 19–26, 2010.

[Zhang *et al.*, 2020] Hanyuan Zhang, Xinyu Zhang, Qize Jiang, et al. Trajectory similarity learning with auxiliary supervision and optimal matching. In *IJCAI*, pages 3209–3215, 2020.

[Zhao *et al.*, 2021] Jiawei Zhao, Wei Luo, Boxing Chen, et al. Mutual-learning improves end-to-end speech translation. In *EMNLP*, pages 3989–3994, 2021.

[Zhou *et al.*, 2018] Fan Zhou, Qiang Gao, Goce Trajcevski, et al. Trajectory-user linking via variational autoencoder. In *IJCAI*, pages 3212–3218, 2018.

[Zhou *et al.*, 2021a] Fan Zhou, Yurou Dai, Qiang Gao, et al. Self-supervised human mobility learning for next location prediction and trajectory classification. *KBS*, page 107214, 2021.

[Zhou *et al.*, 2021b] Fan Zhou, Ruiyang Yin, et al. Improving human mobility identification with trajectory augmentation. *GeoInformatica*, 25(3):453–483, 2021.

[Zuo *et al.*, 2020] Simiao Zuo, Haoming Jiang, Zichong Li, et al. Transformer hawkes process. In *ICML*, pages 11692–11702. PMLR, 2020.