

# Scalable Trajectory-User Linking with Dual-Stream Representation Networks

Hao Zhang<sup>1\*</sup>, Wei Chen<sup>2\*</sup>, Xingyu Zhao<sup>1</sup>, Jianpeng Qi<sup>1</sup>, Guiyuan Jiang<sup>1</sup>, Yanwei Yu<sup>1†</sup>

<sup>1</sup>Faculty of Information Science and Engineering, Ocean University of China, Qingdao, China

<sup>2</sup>The Hong Kong University of Science and Technology (Guangzhou)  
 {zhanghao9222,zhaoxingyu}@stu.ouc.edu.cn, onedeanxxx@gmail.com,  
 {qijianpeng,jiangguiyuan,yuyanwei}@ouc.edu.cn

## Abstract

Trajectory-user linking (TUL) aims to match anonymous trajectories to the most likely users who generated them, offering benefits for a wide range of real-world spatio-temporal applications. However, existing TUL methods are limited by high model complexity and poor learning of the effective representations of trajectories, rendering them ineffective in handling large-scale user trajectory data. In this work, we propose a novel Scalable Trajectory-User Linking with dual-stream representation networks for large-scale TUL problem, named ScaleTUL. Specifically, ScaleTUL generates two views using temporal and spatial augmentations to exploit supervised contrastive learning framework to effectively capture the irregularities of trajectories. In each view, a dual-stream trajectory encoder, consisting of a long-term encoder and a short-term encoder, is designed to learn unified trajectory representations that fuse different temporal-spatial dependencies. Then, a TUL layer is used to associate the trajectories with the corresponding users in the representation space using a two-stage training model. Experimental results on check-in mobility datasets from three real-world cities and the nationwide U.S. demonstrate the superiority of ScaleTUL over state-of-the-art baselines for large-scale TUL tasks.

## Introduction

The proliferation of various location-based service platforms has generated a vast amount of user mobility data, offering researchers valuable opportunities to understand and analyze human movement patterns (Wang, Cao, and Philip 2020). Trajectory-user linking (TUL) aims to associate given anonymous trajectories with the most likely users who generated them (Gao et al. 2017). This problem has garnered significant research interest in recent years because effective TUL methods can benefit a wide range of spatio-temporal applications, including personalized location recommendations (Liu et al. 2019), route suggestions (Pyshkin et al. 2016), behavior tracking (Chen, Hu, and Wang 2019), and the identification of terrorist or criminal activities (Huang et al. 2018). Furthermore, such methods can also be used to test and challenge privacy protection mechanisms in mobility data analysis models (Rao et al. 2020).

\*These authors contributed equally.

†Corresponding author: Yanwei Yu.

In this paper, we focus on addressing the large-scale TUL problem using deep learning models. Traditional methods rely mainly on similarity measurement techniques, such as LCSS (Ying et al. 2010), STS (Li et al. 2021), and SR (Jin et al. 2019, 2020), to assess the similarity between trajectories and link anonymous trajectories with users of similar ones. Early deep learning models typically employ deep neural networks, such as Seq2Seq (Gao et al. 2017), Variational Autoencoders (VAE) (Zhou et al. 2018), and Recurrent Neural Networks (RNNs) (Medsker, Jain et al. 2001), to learn spatiotemporal representations of trajectories and model the problem as a trajectory classification task. Recently, models like MainTUL (Chen et al. 2022) and S<sup>2</sup>TUL (Deng et al. 2023) have been proposed, utilizing mutual distillation learning networks and graph neural networks to capture spatiotemporal movement patterns and relationships between trajectories for TUL problem, respectively. Additionally, a greedy trajectory-user re-linking method has been designed to address the issue of time conflicts in S<sup>2</sup>TUL.

Despite the notable progress achieved by these methods, they still face two major limitations. *First*, existing methods mainly focus on the TUL problem on hundreds of users using trajectory classification methods, while real-world application scenarios are large-scale and involve massive trajectory users. Current deep learning models are not specifically designed for large-scale user trajectory data. They are typically formulated as multi-class trajectory classification problems, which results in inefficiencies and limited performance (Joulin et al. 2017). *Second*, most TUL methods directly learn the representation of trajectories divided by day or longer historical trajectories, lacking effective exploration of the irregularity of trajectories and failing to effectively capture various different spatio-temporal dependencies in large-scale user trajectories.

To address these challenges, we propose ScaleTUL, a Scalable Trajectory-User Linking with dual-stream representation networks for large-scale TUL problem. Specifically, we design a spatio-temporal augmentation strategy that includes both temporal and spatial augmentations. The temporal augmentation introduces trajectories with different time granularities to fully capture user movement patterns across various periods, while the spatial augmentation addresses trajectory sparsity and enhances the potential irreg-

ularities of trajectories to improve model robustness. Additionally, we develop a dual-stream trajectory encoder, consisting of a structured state space model (Gu and Dao 2023) to capture long-term spatio-temporal dependencies in temporal extended trajectories and a recurrent neural network variant (Graves and Graves 2012) to learn short-term spatio-temporal movement patterns. Through dual-view contrastive learning, the model effectively learns the aligned and consistent trajectory representations and the corresponding user label representations. Finally, a two-stage training process is employed to link the trajectories with their corresponding user labels in the representation space.

On three cities and across the nationwide United States check-in dataset from Foursquare, our experiments show that ScaleTUL significantly outperforms state-of-the-art baselines in TUL task.

Our contributions can be summarized as follows:

- We propose a novel scalable trajectory-user linking with dual-stream representation network, named ScaleTUL. Our ScaleTUL effectively and efficiently addresses the challenge of large-scale TUL.
- We design a dual-stream trajectory encoder that includes two distinct encoders, effectively capturing both the long-term and short-term spatio-temporal dependencies of user trajectories, in combination with the spatio-temporal augmentation strategy.
- We conduct extensive experiments on check-in mobility data from three cities as well as across the entire United States. The experimental results demonstrate the effectiveness and superiority of our model in addressing large-scale TUL problem against state-of-the-art baselines.

## Related Work

To address the TUL problem, researchers have developed various approaches, which can be broadly categorized into three groups:

**Similarity measure-based methods.** Traditionally, the TUL problem is addressed through trajectory similarity retrieval methods (Magdy et al. 2015; Sousa, Boukerche, and Loureiro 2020).

Commonly used similarity measures include Longest Common Subsequence (LCSS) (Ying et al. 2010), dynamic time warping (DTW) (Keogh and Pazzani 2000), Signature Representation (SR) (Jin et al. 2019), and spatial-temporal similarity (STS) (Li et al. 2021). *However, due to the complexity of similarity computations, these approaches tend to be computationally expensive and time-consuming.*

**Statistical learning-based methods.** Typically, these methods extract statistical features from trajectory data and train supervised models to predict the likelihood that a user generated a given trajectory (Pao et al. 2012; Ren et al. 2020). For instance, (Ren et al. 2020) employs both profile features and online features derived from trajectory data to train a binary classifier for determining if two trajectories belong to the same user. (Najjar and Mede 2022) is the first work to address the challenge of large-scale TUL by using a non-parametric classifier. *Such models’ effectiveness de-*

*pends on hand-crafted features’ quality, which relies on domain knowledge and understanding of the trajectory dataset.*

**Deep learning-based methods.** Thanks to the powerful learning capabilities of neural networks, numerous deep learning-based methods have been developed to address the TUL problem using check-in trajectory data (Chen et al. 2024a,b). These approaches primarily focus on learning high-order representations of user mobility patterns. For instance, TULER (Gao et al. 2017) employs RNNs like LSTM to learn sequential transition patterns, while TUL-AVE (Zhou et al. 2018) uses a VAE with RNNs to capture hierarchical semantics. DeepTUL (Miao et al. 2020) leverages a bidirectional LSTM with attention for learning periodic mobility patterns, and GAN-based models such as AdattTUL (Gao et al. 2020), TGAN (Zhou et al. 2021), and TULMAL (Zhang et al. 2022) have also been proposed. More recently, MainTUL (Chen et al. 2022) utilizes knowledge distillation with RNN and Transformer encoders, S<sup>2</sup>TUL (Deng et al. 2023) constructs graphs to integrate trajectory-level information, and SAMLink (Chen et al. 2023) designs adaptive fusion modules for different types of trajectory data. *However, previous studies have been limited to classifying trajectories for at most a few hundred users, without addressing scalability problem. This limitation stems from the use of complex neural network architectures combined with classification layers, which are both time-inefficient and memory-intensive.*

## Preliminaries

**Definition 1.** (*Check-in Trajectory*). A check-in trajectory is a chronological sequence of check-in records, where each record is represented as a tuple  $(u, p, c, t)$ . In this tuple,  $u$  represents the user,  $p$  denotes a uniquely identified venue,  $c$  is the category of the venue, and  $t$  is the timestamp of the check-in. The trajectory, denoted as  $Tr_\tau$ , is generated by a user  $u$  within a specific time interval  $\tau$ .

When the user associated with a trajectory is unknown, it is referred to as an unlinked trajectory. In this paper, we define the time interval  $\tau$  as one day. The Trajectory-User Linking (TUL) problem aims to map these unlinked trajectories to their corresponding users.

**Problem.** (*Trajectory-User Linking*). The TUL task aims to link anonymous trajectories to their respective users. Let  $\mathcal{T} = \{Tr^1, Tr^2, \dots, Tr^n\}$  be the set of unlinked trajectories, and  $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$  represent the set of users.

*Our goal is to identify a mapping function  $f(\cdot)$  that links unknown trajectories to the corresponding users:  $\mathcal{T} \mapsto \mathcal{U}$ . This function should map each trajectory to a representation that reflects the user’s identity, ensuring correct association with the rightful user.*

## Methodology

In this section, we present detailed introduction about the trajectory user linking framework based on dual-stream networks (as illustrated in Figure 1). ScaleTUL is accomplished through a two-stage process involving representation learning and user linking. We delve into how to learn high-quality trajectory representations from the perspectives of diverse

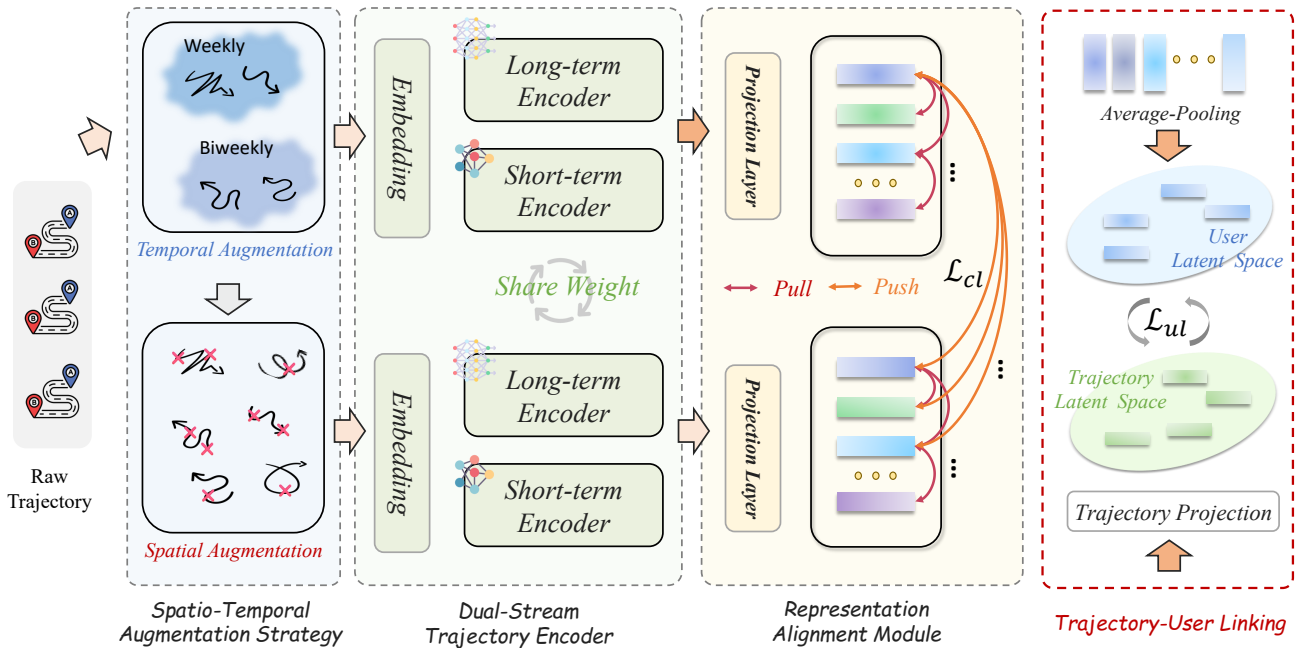


Figure 1: The overview of the proposed framework.

feature capture and lightweight trajectory encoding. Specifically, we first introduce three key components of trajectory representation: *Spatio-Temporal Augmentation Strategy*, *Dual-Stream Trajectory Encoder*, and *Representation Alignment Module*. Then, we introduce how to efficiently link representations in trajectory and user latent spaces.

### Spatio-Temporal Augmentation Strategy

Raw trajectory data of human mobility is often sparse and unable to capture the diverse underlying movement intentions. To address this problem, we propose two augmentation strategies from both temporal and spatial perspectives.

**Temporal Augmentation.** Human mobility information typically exhibits periodic characteristics over time. To fully capture users' temporal visitation preferences and movement patterns across different scales, we propose a periodic temporal augmentation strategy. Specifically, the raw check-in sequences are often segmented into daily trajectories, denoted as  $\{Tr_\tau | \tau = 1\}$ . We then query the trajectory sequences of the current user for the upcoming week and the next two weeks, denoted as  $\{Tr_\tau | 1 \leq \tau \leq 7\}$  and  $\{Tr_\tau | 1 \leq \tau \leq 14\}$ , respectively. Subsequently, the temporal augmented trajectories from different periodic are consolidated into new training samples.

**Spatial Augmentation.** Due to the varying check-in preferences of users across different locations and the potential for communication failures in recording devices, the resulting check-in sequences often exhibit low-sampled trajectories with significant missing locations. To effectively simulate the irregular mobility patterns and noise characteristics that lead to such sparsely sampled movement data, we propose a random spatial augmentation strategy. Specifically, for each newly generated check-in trajectory sample derived

from temporal augmentation, we further apply a random location masking technique to produce a corresponding augmented trajectory, denoted as  $Tr'_\tau = Tr_\tau \odot M_r$ . Here,  $M_r$  represents a binary mask vector, where the masking rate  $r$  is defined as the ratio of the number of masked location points to the total length of the trajectory.

It is worth noting that these two augmentation strategies are applied only during the training phase. By employing these two mobility data augmentation strategies, not only is the issue of sparse training data alleviated from a spatial perspective, but the model is also better able to capture more diverse movement patterns from a temporal perspective.

### Dual-Stream Trajectory Encoder

Due to the design of the augmentation strategies, significant differences in sequence lengths between different trajectories are inevitable, resulting in a multi-modal distribution. Consequently, a simple trajectory encoder is insufficient for modeling such overlapping distribution characteristics. To address this limitation, we propose a dual-stream trajectory encoder, which includes a unified embedding layer along with separate short-term and long-term encoders, each designed to model patterns with different length distributions.

**Unified Embedding Layer.** Each check-in location of trajectory typically includes not only geographic information but also rich external information, such as time and semantic categories. Therefore, before capturing sequential patterns, it is necessary to integrate all the above information into a unified location embedding. Here, we use a simple nonlinear layer fusion to obtain the semantic location representation:

$$x_i = f(\theta; p_i, t_i, c_i), \quad (1)$$

where  $p$ ,  $t$ , and  $c$  denote the discrete mapping of location  $p$  at time  $t$  with category  $c$ . The function  $f_\theta$  is a simple multi-

layer perceptron with a nonlinear activation function.

The complete embedding matrix for a single trajectory can then be obtained by stacking the representations of all individual locations:

$$\mathbf{X} = \{x_i | i = 1, 2, \dots, n\}. \quad (2)$$

By doing so, each trajectory is preliminarily mapped into a high-dimensional semantic representation.

**Dual-Stream Encoder.** As previously mentioned, the augmented trajectories typically exhibit significant differences in sequence lengths. Therefore, it is necessary to design appropriate trajectory encoders for these two different lengths of trajectories to capture their respective features. Additionally, due to the introduction of more trajectory encoders, to maintain efficiency in the sequence learning process, it is essential to select sufficiently lightweight encoding models to support the training of large-scale trajectory data.

To model short-term movement patterns, we select a simple recurrent neural network variant, Bi-directional Long Short-Term Memory (Graves and Graves 2012), as the short-term encoder  $\mathcal{F}_{\theta_\alpha}$ . Furthermore, we use the hidden layer output at the last time step to represent the dynamic pattern of the short trajectory:

$$z_\alpha = \mathcal{F}(\theta_\alpha; \mathbf{X}). \quad (3)$$

Unlike the simple short-term patterns, long-term trajectory patterns require selective processing of input location information while also satisfying long-range dependency modeling. Recently, structured state space models (Gu and Dao 2023) have gained attention due to their hardware-aware lightweight and efficient characteristics, which fulfill our requirements. Therefore, we select them as the long-term encoder  $\mathcal{F}_{\theta_\beta}$ . Furthermore, we use the hidden state at the last time step as the long trajectory representation:

$$z_\beta = \mathcal{F}(\theta_\beta; \mathbf{X}). \quad (4)$$

Then we use a learnable parameter  $\gamma$  to balance the weights of the representations from two encoders to obtain the final representation, and use ridge normalization to standardize the trajectory representation, improving numerical stability and accelerating the model’s training process:

$$z = \|(\gamma \cdot z_\alpha + (1 - \gamma) \cdot z_\beta)\|_2. \quad (5)$$

## Representation Alignment Module

Intuitively, by applying spatial augmentation strategies, we obtain two views of trajectory samples. A natural approach is to align the masked-view trajectory with the unmasked-view trajectory via contrastive learning to ensure robust trajectory representation (Li et al. 2023). However, unlike traditional self-supervised learning, which relies on contrastive learning with negative samples, our unique advantage lies in the accessibility of trajectory labels. Therefore, in addition to basic negative sample information, we can also consider the associated positive sample signals for each anchor (in contrast to traditional self-supervised contrastive learning that typically uses only a single positive sample signal). By leveraging labeled data to enhance contrastive learning, we

can obtain higher-quality trajectory representations, thereby avoiding the misclassification of the same user’s trajectories as negative samples.

Specifically, given a batch of  $N$  unmasked random sample-label pairs, the corresponding  $2N$  multi-view sample pairs are generated through spatial masking augmentation strategies. We then apply an extended supervised version (Khosla et al. 2020) of the traditional Noise Contrastive Estimation loss (InfoNCE) (Chen et al. 2020) to achieve more effective representation alignment:

$$\mathcal{L}_{cl} = \sum_{i \in 2N} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \eta)}{\sum_{b \in B(i)} \exp(z_i \cdot z_b / \eta)}, \quad (6)$$

where  $z_* = \|Proj(z)\|_2$  means that the representation space is mapped to the unit hypersphere through multi-layer perceptron and ridge regularization, so that the similarity can be measured by the inner product.  $\eta \in \mathcal{R}^+$  means a scalar temperature parameter.  $i$  is the index of anchor.  $B(i) \equiv \{1, \dots, 2N\} \setminus \{i\}$  is the set of indices of all samples among the  $2N$  trajectories, excluding the index  $i$ .  $P(i) \equiv \{p \in B(i) : Label(p) = Label(i)\}$  is the set of indices of all positive samples among the  $2N$  trajectories.

By completing contrastive learning accompanied by supervision signals, we effectively align a more discriminative representation space, where the trajectory representations of the same user are closer, and those of different users are further apart. Subsequently, we discard the projection layer, freeze the parameters of the embedding layer and dual-stream encoder  $\mathcal{F}_{\theta_\alpha}$  and  $\mathcal{F}_{\theta_\beta}$ , and input all the trajectories of each user in the training set into the backbone network to obtain the individual trajectory representations:

$$z^\Delta = \|(\gamma \cdot \mathcal{F}(\theta_\alpha; \mathbf{X}) + (1 - \gamma) \cdot \mathcal{F}(\theta_\beta; \mathbf{X}))\|_2. \quad (7)$$

Considering that traditional supervised loss typically uses a fixed cross-entropy loss classifier, it does not scale well to dynamic and large-scale trajectory-user linking. Therefore, we propose using a cosine similarity loss to avoid this issue. Specifically, by aggregating the trajectory representations of a single user through a pooling layer, we can precompute the user labels needed for the next stage of training:

$$u_i = \text{Average-Pooling}(\{z^j | j \in U_i\}), \quad (8)$$

where  $z^j$  means all trajectory representations of user  $U_i$ .

## Trajectory-User Linking Layer

To link unknown trajectories to the users who generated them, we further designed a TUL (Trajectory-User Link) predictor, composed of activation functions and a multi-layer perceptron, to obtain the final trajectory representation  $y = f(\theta; z)$ . This projection layer is intended to further enhance the representational capacity of the input trajectories.

In this representation space, we minimize the cosine similarity loss between the representation of the input trajectory and the corresponding user representation:

$$\mathcal{L}_{ul} = \sum_{i \in N} (1 - \cos(y_i, u_i)), \quad (9)$$

Datasets	#users	#trajectories	#POIs	#categories	Duration
TKY	800	104,413	39,698	239	11 months
	400	51,969	24,526	231	11 months
NYC	800	152,583	31,925	251	10 months
	400	39,784	19,903	248	10 months
Santiago	800	132,238	21,967	381	18 months
	400	64,932	13,703	357	18 months
US	5000	749,443	258,865	427	18 months
	10000	1,167,786	344,274	428	18 months

Table 1: Statistics of the datasets.

where  $u_i$  denotes the user label who generated trajectory  $i$ ,  $y_i$  means unlinked trajectory representation,  $N$  is number of the trajectories. During the testing stage, by calculating the similarity between the input trajectories and the corresponding user representations, each input trajectory is linked with the most similar user label.

## Model Training

Our model uses a two-stage training pipeline to capture users’ mobility characteristics and learn generalized features. By integrating semantic information from both raw and augmented trajectories, we obtain enriched user and trajectory representations.

In the first stage, the input trajectory is augmented, and both the input and augmented trajectories are passed through the dual-stream encoder separately to obtain their respective latent features. These features are then projected into a lower-dimensional space via a projection layer, where they are normalized and used to compute the contrastive learning loss. In the second stage, we train trajectory-user linking layer by minimizing the cosine similarity loss between the representations of the trajectory and the corresponding user.

## Experiments

### Datasets

We use real-world check-in data collected from the popular location-based social network platform Foursquare (Yang et al. 2015), selecting data from three cities and the entire United States for our dataset. For the cities of Tokyo (TKY), New York City (NYC), and Santiago, we select the top 800 and 400 users, respectively, who have the most trajectories, to evaluate model performance. To demonstrate the effectiveness of our proposed framework on a large scale, we select the top 5000 and 10000 users with the most trajectories across the United States for evaluation. In our experiments, we use the first 80% of each user’s sub-trajectories for training, and the remaining 20% for testing. Additionally, 20% of the training data is set aside as a validation set to assist with an early stopping mechanism to find the best parameters and avoid overfitting. The statistics of all datasets are summarized in Table 1.

### Approach for Comparison

We compare ScaleTUL with recent state-of-the-art models for trajectory user-linking. Specifically, the following baseline approaches are evaluated.

- **KNNTUL** (Najjar and Mede 2022) - is the first work to attempt solving the large-scale TUL problem by employing the k-Nearest Neighbors (KNN) method.
- **MainTUL** (Chen et al. 2022) - is a knowledge distillation framework to capture rich contextual features in check-in data, with an RNN encoder and a temporal-aware Transformer encoder.
- **S<sup>2</sup>TUL** (Deng et al. 2023) - is a semi-supervised framework that models the relationships between trajectories by constructing both homogeneous and heterogeneous graphs. The framework includes four variants, of which we select the two that performed best across most datasets: S<sup>2</sup>TUL-R and S<sup>2</sup>TUL-HRSTS.

### Evaluation Metrics and Parameter Settings

We use the Acc@k, Macro-F1 and Micro-F1 to evaluate the model performance. Specifically, Acc@k is used to evaluate the accuracy of TUL prediction. Unlike previous work that used Macro-Precision and Macro-Recall, we opted to include Micro-F1 for the following reasons: Macro-F1 is derived from Macro-Precision and Macro-Recall, effectively integrating their results. Additionally, Micro-F1 offers an overall performance metric that more accurately reflects the model’s behavior across the entire dataset. For the baselines, we use the parameter settings recommended in their papers and fine-tune them. For ScaleTUL, we set default embedding dimension to 512, apply an early stopping mechanism with patience to 5 to avoid over fitting, and adjust the learning rates as follows: In the first stage, the initial learning rate is set to 0.001 and decays by 20% every 5 epochs. In the second stage, the initial learning rate is set to 0.0005 and decays by 90% every 5 epochs. The source code of our model is available at <https://github.com/sleevefishcode/ScalableTUL>.

All experiments are conducted on a machine with Intel(R) Xeon(R) Silver 4214 (2.20GHz 12 cores) and NVIDIA GeForce RTX 3090 (24GB Memory).

### Overall Performance

We report the overall performance with baselines in Table 2 and Table 3, where the best is shown in bold and the second best is shown as underlined.

As shown in Table 2, ScaleTUL significantly outperforms all baselines in terms of all evaluation metrics on the check-in datasets of three cities, except for Acc@5 on TKY. This performance can be attributed to the spatio-temporal augmentation strategy and the dual-stream trajectory encoder. The former helps to alleviate issues related to trajectory irregularity and sparsity, thereby enhancing the model’s generalization capability, while the latter more effectively captures the spatio-temporal movement patterns in users’ check-in trajectories, accommodating different trajectory lengths.

Dataset	Methods	Acc@1	Acc@5	Macro-F1	Micro-F1	Acc@1	Acc@5	Macro-F1	Micro-F1
		$ \mathcal{U}  = 400$				$ \mathcal{U}  = 800$			
TKY	KNNTUL	44.32%	57.13%	43.76%	44.32%	37.45%	50.02%	36.21%	37.46%
	MainTUL	60.69%	69.80%	59.18%	60.69%	<u>57.02%</u>	<u>67.82%</u>	<u>55.48%</u>	<u>57.02%</u>
	S <sup>2</sup> TUL-R	59.98%	73.04%	60.82%	59.98%	OOM	OOM	OOM	OOM
	S <sup>2</sup> TUL-HRSTS	63.53%	<b>75.90%</b>	62.94%	63.71%	OOM	OOM	OOM	OOM
	<b>ScaleTUL</b>	<b>65.13%*</b>	<u>74.47%*</u>	<b>63.37%*</b>	<b>65.13%*</b>	<b>60.63%*</b>	<b>71.03%*</b>	<b>58.74%*</b>	<b>60.63%*</b>
NYC	KNNTUL	44.69%	54.31%	43.81%	44.68%	39.15%	49.95%	37.70%	39.15%
	MainTUL	54.84%	60.83%	53.36%	54.83%	<u>52.43%</u>	59.50%	<u>51.00%</u>	<u>52.43%</u>
	S <sup>2</sup> TUL-R	57.00%	64.35%	<u>56.52%</u>	<u>57.00%</u>	51.97%	<u>60.52%</u>	50.56%	51.96%
	S <sup>2</sup> TUL-HRSTS	56.88%	<u>65.52%</u>	54.65%	56.88%	51.59%	60.39%	49.00%	51.58%
	<b>ScaleTUL</b>	<b>59.43%*</b>	<b>66.03%*</b>	<b>57.06%*</b>	<b>59.43%*</b>	<b>56.11%*</b>	<b>63.78%*</b>	<b>53.30%*</b>	<b>56.11%*</b>
Santiago	KNNTUL	52.35%	62.78%	53.39%	52.34%	46.56%	57.41%	47.74%	46.57%
	MainTUL	55.29%	64.59%	55.44%	55.29%	<u>51.82%</u>	<u>63.00%</u>	51.61%	<u>51.82%</u>
	S <sup>2</sup> TUL-R	53.96%	<u>64.95%</u>	54.70%	53.96%	49.46%	61.40%	50.08%	49.46%
	S <sup>2</sup> TUL-HRSTS	54.00%	64.10%	53.34%	54.00%	OOM	OOM	OOM	OOM
	<b>ScaleTUL</b>	<b>58.44%*</b>	<b>67.12%*</b>	<b>57.46%*</b>	<b>58.43%*</b>	<b>53.55%*</b>	<b>64.51%*</b>	<b>52.61%*</b>	<b>53.55%*</b>

Table 2: Performance comparison with baselines on the dataset from three cities. OOM stands for “out of memory”. Marker \* indicates the results are statistically significant (t-test with p-value < 0.01).

Dataset	Methods	Acc@1	Acc@5	Macro-F1	Micro-F1	Acc@1	Acc@5	Macro-F1	Micro-F1
		$ \mathcal{U}  = 5000$				$ \mathcal{U}  = 10000$			
US	KNNTUL	45.59%	57.36%	44.85%	45.59%	39.00%	<u>51.13%</u>	37.70%	39.01%
	MainTUL	<u>51.40%</u>	<u>57.71%</u>	<u>50.89%</u>	<u>51.39%</u>	<u>43.85%</u>	51.01%	<u>42.83%</u>	<u>43.85%</u>
	S <sup>2</sup> TUL-R	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	S <sup>2</sup> TUL-HRSTS	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	<b>ScaleTUL</b>	<b>54.94%*</b>	<b>64.05%*</b>	<b>54.02%*</b>	<b>54.94%*</b>	<b>44.90%*</b>	<b>53.88%*</b>	<b>44.32%*</b>	<b>44.90%*</b>

Table 3: Performance comparison with baselines on the dataset from the United States. OOM stands for “out of memory”. Marker \* indicates the results are statistically significant (t-test with p-value < 0.01).

As the number of users in a dataset increases, the challenge of correctly linking users to their trajectories intensifies, leading to performance degradation in all models. However, ScaleTUL can still achieve the best results. Specifically, ScaleTUL achieves an average improvement of 6.68% in Acc@1 and 5.2% in Macro-F1 compared to the best-performing baseline on the TKY and NYC datasets when the user count reaches 800. The significant improvements in accuracy and F1 scores suggest that ScaleTUL’s architecture, especially its spatio-temporal trajectory augmentation strategy and dual-stream encoder, effectively captures complex user mobility patterns even as the number of users scales up, demonstrating its robustness against the increased complexity of larger datasets.

The experimental results on the nationwide U.S. dataset are presented in Table 3. Notably, ScaleTUL continues to perform well even when the number of users reaches 10k. In contrast, S<sup>2</sup>TUL faces challenges in managing the large volume of user-generated trajectories, and the performance of the KNNTUL method falls significantly short of our model’s performance. For 5k users, our model surpasses the previ-

ously best-performing MainTUL model, with improvements of 6.89% in Acc@1 and 6.11% in Macro-F1.

The consistent outperformance of ScaleTUL across different user scales not only underscores its robustness but also demonstrates its scalability. The model’s ability to maintain and even improve its performance as the dataset grows larger suggests that its architecture is well-suited for handling the increased complexity and data irregularities that come with larger user bases. These significant improvements emphasize the superiority of our approach in effectively learning and generalizing from diverse and extensive trajectory data, making ScaleTUL a reliable and powerful solution for large-scale TUL applications.

### Ablation Study

In our ablation study, we evaluate our model against five carefully designed variations: (1) **w/o T-Aug** – This variant excludes temporal augmentation from the trajectory augmentation strategy. (2) **w/o S-Aug** – This variation omits the random mask used in trajectory augmentation. (3) **w/o Aug** – Here, the entire trajectory augmentation strategy is removed,

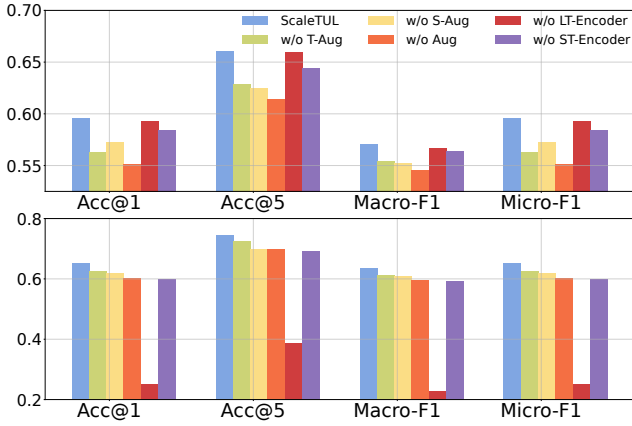


Figure 2: Experimental results of component ablation on NYC (top) and TKY (bottom) ( $|\mathcal{U}| = 400$ ).

meaning no aggregation or masking is applied. (4) **w/o LT-Encoder** – It removes the long-term encoder in dual-stream trajectory encoder. (5) **w/o ST-Encoder** – It excludes the short-term encoder in dual-stream trajectory encoder.

The results of the ablation study are shown in Figure 2, which demonstrates the contribution of each component to the overall performance of our model. Notably, the long-term encoder has a significant affect on model performance, especially on the TKY dataset. This is primarily due to the fact that the TKY dataset includes a substantially higher number of longer trajectories (those with five or more points) compared to the NYC dataset. This highlights the effectiveness of our designed trajectory encoder, which includes the long-term component. The impact of the trajectory augmentation strategy—**w/o T-Aug**, **w/o S-Aug**, and **w/o Aug**—underscores the importance. These variations reveal that temporal and spatial trajectory augmentations are crucial for improving model performance, particularly in handling data irregularities and sparsity. This is particularly evident in the NYC dataset, where trajectory augmentation helps mitigate issues related to data sparsity and irregularity, demonstrating its effectiveness in enhancing the model’s ability to generalize from limited data.

### Parameter Study

We evaluate the model’s sensitivity to embedding dimension  $d$  and masking rate  $r$ . As shown in Figure 3, the model’s

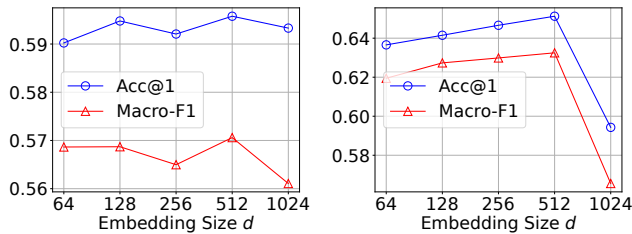


Figure 3: Result of parameter sensitivity embedding size  $d$  on NYC(left) and TKY(right) ( $|\mathcal{U}| = 400$ ).

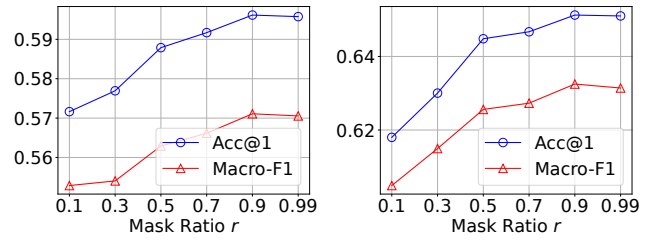


Figure 4: Result of parameter sensitivity mask ratio  $r$  on NYC(left) and TKY(right) ( $|\mathcal{U}| = 400$ ).

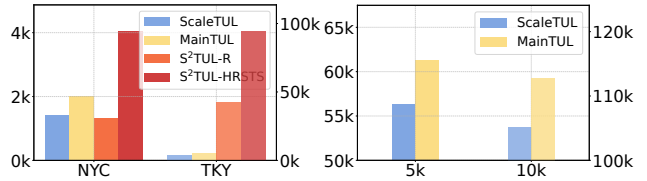


Figure 5: Experimental results of running time,  $|\mathcal{U}| = 400$  (left) and  $|\mathcal{U}| = 5000/10000$  (right).

performance initially improves with  $d$  increases and then declines. This trend occurs because a small embedding dimension is insufficient to capture the spatio-temporal dependencies intrinsic to trajectories, while an overly large embedding dimension reduces the model’s generalization capability. For masking rate  $r$ , larger values improve performance by alleviating data sparsity and boosting the model’s ability to generalize. However, when  $r$  approaches 1, performance plateaus and eventually declines, as excessive masking eliminates important information that the model needs to make accurate predictions.

### Efficiency Study

We also perform efficiency experiments by comparing our model with the baseline models. As shown in Figure 5, the results indicate that our model outperforms other deep learning methods in terms of time efficiency. Particularly on the TKY dataset, our model achieve a speed improvement of approximately 96.04% compared to the most effective baseline,  $S^2$ TUL-HRSTS. The reason for its significant time consumption is that constructing both homogeneous and heterogeneous graphs to integrate trajectory-level information becomes highly time-consuming when dealing with a large number of trajectories. It should be noted that when the number of users is 10k, the number of trainable parameters is reduced by 10 million compared to MainTUL, thereby lowering hardware demands and shortening training time.

### Conclusion

In this paper, we propose a novel scalable trajectory-user linking with dual-stream representation networks called ScaleTUL. Experimental results on datasets from three cities and across the United States demonstrate that ScaleTUL outperforms the state-of-the-art methods and proves its efficiency in efficiency study.

## Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant Nos. 62176243 and 62372421, the Fundamental Research Funds for the Central Universities under Grant No 202442005, the National Key R&D Program of China under Grant No 2022ZD0117201, and Shandong Postdoctoral Innovative Talent Support Program under Grant No. SDBX2023013.

## References

- Chen, C.-X.; Zhang, W.; Yu, B.; and Chen, C. 2023. SAMLink: a mobility signature augmentation model for trajectory-user linking. *Neural Computing and Applications*, 35(34): 24473–24491.
- Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.
- Chen, W.; Huang, C.; Yu, Y.; Jiang, Y.; and Dong, J. 2024a. Trajectory-User Linking via Hierarchical Spatio-Temporal Attention Networks. *ACM Trans. Knowl. Discov. Data*, 18(4).
- Chen, W.; Li, S.; Huang, C.; Yu, Y.; Jiang, Y.; and Dong, Y. 2022. Mutual Distillation Learning Network for Trajectory-User Linking. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.
- Chen, W.; Liang, Y.; Zhu, Y.; Chang, Y.; Luo, K.; Wen, H.; Li, L.; Yu, Y.; Wen, Q.; Chen, C.; et al. 2024b. Deep learning for trajectory data management and mining: A survey and beyond. *arXiv preprint arXiv:2403.14151*.
- Chen, Y.; Hu, C.; and Wang, J. 2019. Human-centered trajectory tracking control for autonomous vehicles with driver cut-in behavior prediction. *IEEE Transactions on Vehicular Technology*, 68(9): 8461–8471.
- Deng, L.; Sun, H.; Zhao, Y.; Liu, S.; and Zheng, K. 2023. S2tul: A semi-supervised framework for trajectory-user linking. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, 375–383.
- Gao, Q.; Zhang, F.; Yao, F.; Li, A.; Mei, L.; and Zhou, F. 2020. Adversarial mobility learning for human trajectory classification. *IEEE Access*, 8: 20563–20576.
- Gao, Q.; Zhou, F.; Zhang, K.; Trajcevski, G.; Luo, X.; and Zhang, F. 2017. Identifying Human Mobility via Trajectory Embeddings. In *IJCAI*, volume 17, 1689–1695.
- Graves, A.; and Graves, A. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37–45.
- Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Huang, C.; Zhang, J.; Zheng, Y.; and Chawla, N. V. 2018. DeepCrime: Attentive hierarchical recurrent networks for crime prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 1423–1432.
- Jin, F.; Hua, W.; Xu, J.; and Zhou, X. 2019. Moving object linking based on historical trace. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 1058–1069. IEEE.
- Jin, F.; Hua, W.; Zhou, T.; Xu, J.; Francia, M.; Orłowska, M. E.; and Zhou, X. 2020. Trajectory-based spatiotemporal entity linking. *IEEE Transactions on Knowledge and Data Engineering*, 34(9): 4499–4513.
- Joulin, A.; Cissé, M.; Grangier, D.; Jégou, H.; et al. 2017. Efficient softmax approximation for GPUs. In *International conference on machine learning*, 1302–1310. PMLR.
- Keogh, E. J.; and Pazzani, M. J. 2000. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 285–289.
- Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33: 18661–18673.
- Li, G.; Hung, C.-C.; Liu, M.; Pan, L.; Peng, W.-C.; and Chan, S.-H. G. 2021. Spatial-temporal similarity for trajectories with location noise and sporadic sampling. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 1224–1235. IEEE.
- Li, S.; Chen, W.; Yan, B.; Li, Z.; Zhu, S.; and Yu, Y. 2023. Self-supervised contrastive representation learning for large-scale trajectories. *Future Generation Computer Systems*, 148: 357–366.
- Liu, W.; Wang, Z.-J.; Yao, B.; and Yin, J. 2019. Geo-ALM: POI Recommendation by Fusing Geographical Information and Adversarial Learning Mechanism. In *IJCAI*, volume 7, 1807–1813.
- Magdy, N.; Sakr, M. A.; Mostafa, T.; and El-Bahnasy, K. 2015. Review on trajectory similarity measures. In *2015 IEEE seventh international conference on Intelligent Computing and Information Systems (ICICIS)*, 613–619. IEEE.
- Medsker, L. R.; Jain, L.; et al. 2001. Recurrent neural networks. *Design and Applications*, 5(64-67): 2.
- Miao, C.; Wang, J.; Yu, H.; Zhang, W.; and Qi, Y. 2020. Trajectory-user linking with attentive recurrent network. In *Proceedings of the 19th international conference on autonomous agents and multiagent systems*, 878–886.
- Najjar, A.; and Mede, K. 2022. Trajectory-user linking is easier than you think. In *2022 IEEE International Conference on Big Data (Big Data)*, 4936–4943. IEEE.
- Pao, H.-K.; Fadlil, J.; Lin, H.-Y.; and Chen, K.-T. 2012. Trajectory analysis for user verification and recognition. *Knowledge-Based Systems*, 34: 81–90.
- Pyshkin, E.; Baratynskiy, A.; Chisler, A.; and Skripal, B. 2016. Information management for travelers: Towards better route and leisure suggestion. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 429–438.
- Rao, J.; Gao, S.; Kang, Y.; and Huang, Q. 2020. LSTM-TrajGAN: A deep learning approach to trajectory privacy protection. *arXiv preprint arXiv:2006.10521*.

- Ren, H.; Pan, M.; Li, Y.; Zhou, X.; and Luo, J. 2020. St-siamesenet: Spatio-temporal siamese networks for human mobility identification. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 1306–1315.
- Sousa, R. S. D.; Boukerche, A.; and Loureiro, A. A. 2020. Vehicle trajectory similarity: Models, methods, and applications. *ACM Computing Surveys (CSUR)*, 53(5): 1–32.
- Wang, S.; Cao, J.; and Philip, S. Y. 2020. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, 34(8): 3681–3700.
- Yang, D.; Zhang, D.; Zheng, V. W.; and Yu, Z. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1): 129–142.
- Ying, J. J.-C.; Lu, E. H.-C.; Lee, W.-C.; Weng, T.-C.; and Tseng, V. S. 2010. Mining user similarity from semantic trajectories. In *Proceedings of the 2nd acm sigspatial international workshop on location based social networks*, 19–26.
- Zhang, S.; Wang, S.; Wang, X.; Zhang, S.; Miao, H.; and Zhu, J. 2022. Multi-task adversarial learning for semi-supervised trajectory-user linking. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 418–434. Springer.
- Zhou, F.; Gao, Q.; Trajcevski, G.; Zhang, K.; Zhong, T.; and Zhang, F. 2018. Trajectory-User Linking via Variational AutoEncoder. In *IJCAI*, 3212–3218.
- Zhou, F.; Yin, R.; Trajcevski, G.; Zhang, K.; Wu, J.; and Khokhar, A. 2021. Improving human mobility identification with trajectory augmentation. *GeoInformatica*, 25: 453–483.