Independent University **Bangladesh (IUB)**

IUB Academic Repository

Graduate Thesis 2024

2025-05-13

Trajectory User Linking via TrajTRoPE

Jubaer, Ezharuddin

IUB

http://ar.iub.edu.bd/handle/11348/978

Downloaded from IUB Academic Repository



Trajectory User Linking via TrajTRoPE

May 2025

Prepared by:

Ezharuddin Jubaer
ID: 2120117

Sumaiya Karim Katha
ID: 2010189

Md Fahim Shahriar
ID: 2120552

Department of Computer Science and Engineering
Independent University, Bangladesh

Supervised by

Amin Ahsan Ali, PhD
Professor
Department of Computer Science and Engineering
Independent University, Bangladesh

Attestation

We are aware of the fact that plagiarism is strictly prohibited and it conflicts with our university's rules and regulations. We guarantee the authenticity of our work. We have used some copyrighted material and models of others in our project which has been properly cited following the international standards proper guidelines.

Author Name:

. . . . . .

Signature:

. . . . . .

Author Name:

. . . . . .

Signature:

. . . . . .

Author Name:

. . . . . .

Signature:

. . . . . .

Evaluation Committee

Supervisor

Name: _____ Signature: _____

Internal Examiner 1

Name: _____ Signature: _____

Internal Examiner 2

Name: _____ Signature: _____

External Examiner

Name: _____ Signature: _____

Acknowledgement

We would like to express sincere gratitude to our respected Professor, Amin Ahsan Ali, Independent University, Bangladesh. His dedicated guidance, insightful suggestions, remarkable patience, and the generous investment of his valuable time played a pivotal role in the successful completion of our project. Our Professor's expertise and mentorship greatly enhanced the quality and depth of our work, and we are truly appreciative of his unwavering support throughout our academic journey.

Additionally, our heartfelt thanks go to the CCDS Lab (Center for Computational & Data Sciences) for their extraordinary help and consistent support. The collaborative and resourceful environment provided by CCDS Lab significantly contributed to the development and execution of our project. Moreover, The encouragement and assistance have been instrumental in shaping our academic and research field. We are grateful for the opportunities and resources extended to us during our research, which undoubtedly enriched our learning experience.

We acknowledge and appreciate the contributions of all individuals who supported us, both directly and indirectly, during this endeavor.

Abstract

Trajectory-user linking (TUL) is a critical problem in spatio-temporal data mining that involves associating anonymous trajectories with the users who generated them. In this thesis, we address the TUL challenge by proposing a Transformer-based model augmented with Rotary Positional Embedding (RoPE) encoder and enriched with semantic context through Points of Interest (POI) features. Using the GeoLife dataset as the experimental platform, we conduct extensive evaluations on user subsets with trajectory lengths constrained to a maximum of 1000 points.

Our findings demonstrate that the incorporation of POI features substantially enhances the model's performance, with the full model achieving an Accuracy@1 of 68.33% and a Macro F1 score of 51.34%. Ablation studies reveal that removing either the Transformer structure or the RoPE encoder significantly degrades performance, highlighting the importance of both sequential modeling and positional encoding. Furthermore, feature importance analysis shows that the absence of POI information results in a dramatic collapse of predictive accuracy and generalization, reaffirming the necessity of contextual features for effective user discrimination.

While the model achieves strong results, the study also acknowledges limitations, including reliance on static spatial features and a relatively narrow evaluation scope. Ethical considerations regarding user privacy are discussed, emphasizing the need for responsible data handling and privacy-preserving extensions. Overall, this thesis advances the trajectory-user linking field by demonstrating the synergistic role of structural modeling and semantic enrichment, while paving the way for future research in robust, scalable, and ethically aligned mobility analytics.

Contents

1 Introduction **9**
- 1.1 Background of the project . . . 9
- 1.2 Preliminaries . . . 10
  - 1.2.1 Spatio-Temporal Point . . . 10
  - 1.2.2 Trajectory . . . 10
  - 1.2.3 Map and Points of Interest . . . 10
- 1.3 Problem Statement . . . 11
- 1.4 Outcome of the project . . . 12

2 Literature Review **13**
- 2.1 Existing TUL Models . . . 13
  - 2.1.1 RNN-Based . . . 13
  - 2.1.2 Graph Based . . . 14

3 Dataset Description & Preprocessing **17**
- 3.1 Introduction . . . 17
- 3.2 Structure and Features of Dataset . . . 17
- 3.3 Preprocessing of the Data . . . 18
  - 3.3.1 Trip Preprocessing . . . 18
  - 3.3.2 POI Preprocessing . . . 20

4 Methodology **22**
- 4.1 Overview . . . 22
- 4.2 Transformer . . . 22
  - 4.2.1 Trajectory Modalities and Embeddings . . . 22
  - 4.2.2 Trajectory Modality Mixing . . . 23
- 4.3 STRPE: Spatio-Temporal Rotary Position Embedding . . . 23
- 4.4 Trajectory User Linking (TUL) . . . 24
  - 4.4.1 Input Representation . . . 24
  - 4.4.2 Encoding and Pooling . . . 24
  - 4.4.3 User Prediction Head . . . 25
  - 4.4.4 Training Objective . . . 25

5 Experimental Design **26**
- 5.1 Experimental Settings . . . 26
- 5.2 Evaluation Metrics . . . 26
  - 5.2.1 Accuracy at 1 ($ACC@1$) . . . 27
  - 5.2.2 Accuracy at 5 ($ACC@5$) . . . 27
  - 5.2.3 Macro Precision ($MacroP$) . . . 27
  - 5.2.4 Macro Recall ($MacroR$) . . . 27

5.2.5 Macro F1 Score ($MacroF1$) 28

6 Results and Analysis **29**
6.1 Performance Comparison 29
6.1.1 Setting 1: User Count 89 (Trajectory Length $\leq$ 1000) 29
6.1.2 Setting 2: User Count 56 (Trajectory Length $\leq$ 1000) 29
6.1.3 Analysis and Insights 29
6.2 Ablation Study 31
6.2.1 Full Model (Transformer with RoPE Encoder) 31
6.2.2 Transformer Only (without RoPE Encoder) 31
6.2.3 Importance of RoPE Encoder 34
6.3 Feature Importance 34
6.3.1 Model Performance with POI Features 34
6.3.2 Model Performance without POI Features 34
6.3.3 Importance of POI 36
6.4 Impact of Tokenization on Model Performance 37
6.4.1 Model Performance with Tokens 37
6.4.2 Model Performance without Tokens 37
6.4.3 Analysis 37
6.5 Ablation Summary and Visualization 40

7 Conclusion and Future Works **41**
7.1 Conclusion 41
7.2 Limitations 41
7.3 Ethical Considerations 41
7.4 Impact on Society 42
7.5 Future Work 42

Bibliography **42**

List of Tables

1.1 Notations & Definitions [3] 11

2.1 Summary of Various TUL Models 16

3.1 Finalized Dataset Columns after Trip Preprocessing 20

5.1 Experimental Settings for Trajectory-User Linking 26

6.1 Performance Comparison Between TrajTRoPE and AttnTUL for Users 89 and 56 31

6.2 Unified Performance Comparison: Full Model vs. Transformer Only (No RoPE), Without POI, and Without Tokenization 40

List of Figures

1 Distance (m) between 2 consecutive points 18
2 Trajectory Count per User 19
3 Average Trajectory Length Point Count for 182 users 19

4 Diagram of the pipeline used in this work 23

5 Train and Validation Loss Curves. 30
6 Performance Metrics Over Training Steps for Full Model: (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score. 32
7 Performance Metrics Over Training Steps for Transformer-only Variant: (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score. 33
8 Performance Metrics With POI : (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score. 35
9 Performance Metrics Without POI : (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score. 36
10 Performance Metrics Over Training Steps With Tokens : (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score. 38
11 Performance Metrics Over Training Steps Without Tokens : (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score. 39
12 Bar chart visualization of ablation results for different architectural and feature configurations. 40

1 Introduction

1.1 Background of the project

With the proliferation of GPS-enabled devices and the widespread adoption of location-based services, the volume of mobility data collected from individuals has grown exponentially. This data, typically represented as spatio-temporal trajectories, captures the movement patterns of users through sequences of timestamped geographic locations. The increasing availability of such rich data has opened up a broad spectrum of opportunities in spatial-temporal data mining. Applications range from personalized location-based recommendations and intelligent transportation systems to urban planning, anomaly detection, and behavioral user profiling. Within this evolving landscape, one critical and emerging challenge is Trajectory-User Linking (TUL), the task of associating anonymous trajectories with the most likely users who generated them. This problem is particularly relevant in scenarios where privacy concerns necessitate user anonymization, but practical applications require re-identification or user-specific insights for improved services or security. Early approaches to TUL primarily employed models based on Recurrent Neural Networks (RNNs) [10] and Variational Autoencoders (VAEs) [32] to learn latent representations of users' mobility behaviors. More recently, techniques such as contrastive learning have also been introduced to refine trajectory embeddings by considering spatio-temporal constraints. However, despite their advancements, these models often struggle with key challenges: they are limited in capturing long-range temporal dependencies, they do not generalize well under data sparsity (especially with low-sampling or inactive users), and they typically fall short in encoding both local transitions and global spatial hierarchies present in real-world mobility patterns [3]. To address these shortcomings, AttnTUL was introduced, a hierarchical spatio-temporal attention network designed to overcome the limitations of prior work [3]. AttnTUL integrates both local and global trajectory modeling using graph neural networks and multi-head self-attention, making it particularly effective in learning user-discriminative mobility patterns [3]. The model has demonstrated robust performance on benchmark datasets such as GeoLife [30] and Gowalla [5], which contain both dense GPS and sparse check-in trajectories. AttnTUL's design explicitly captures intra-trajectory sequential dependencies and inter-trajectory spatial relationships, thereby offering a more comprehensive solution to the TUL task [3].

Motivated by the growing effectiveness of Transformer architectures across various sequence modeling tasks, we sought to investigate how a Transformer-based model augmented with Rotary Positional Encoding (RoPE) performs in the context of trajectory-user linking (TUL) based off of TrajFM [7]. While Transformer models have been widely adopted in domains such as natural language processing and time-series forecasting, their application to TUL remains relatively limited. In particular, the use of RoPE as a positional encoding strategy has not been thoroughly explored within this problem space. This study aims to evaluate the capability of such a model to capture both spatial and temporal dependencies in mobility data and determine if it can outperform existing mod-

els.

1.2 Preliminaries

To facilitate a rigorous formulation of the TUL problem, this section introduces the fundamental concepts and notations used throughout this work. These definitions establish the formal framework upon which the subsequent methodology is built.

Let

$$U = \{u_1, u_2, \ldots, u_i\}$$

denote a set of moving users. Each user generates a series of location records over time, referred to as a trajectory. A trajectory consists of discrete points, each associated with both spatial and temporal information.

1.2.1 Spatio-Temporal Point

A **spatio-temporal point** is defined as a unique data entry in the form of a tuple

$$\langle t, \ell \rangle$$

where t is the timestamp indicating when the point was recorded, and ℓ is the geographical coordinate representing the location, typically specified by latitude and longitude.

1.2.2 Trajectory

A **trajectory** is a chronologically ordered sequence of spatio-temporal points generated by a user u_i over a specific time interval τ. It is denoted by

$$Tr^{\tau}_{u_i} = \{\langle t_1, \ell_1 \rangle, \langle t_2, \ell_2 \rangle, \ldots, \langle t_m, \ell_m \rangle\}$$

The time interval τ may vary in granularity, such as an hour, a day, a week, or even a month. If the identity of the user who generated a trajectory is unknown, the trajectory is referred to as *unlinked*. Let T denote the full observation period. The collection of all known or *linked* trajectories generated by all users over previous intervals is denoted by

$$Tr_u = \{Tr^{\tau}_{u_i} \mid u_i \in U, \tau \in T\}$$

1.2.3 Map and Points of Interest

Let M represent a **map**, defined over a finite, continuous geographic space. Within this space, we define a set of **points of interest (POIs)** as

$$P = \{p_1, p_2, \ldots, p_{|P|}\}$$

Each POI p_i corresponds to a distinct location or venue on the map M, such as a restaurant, school, park, or public building.

1.3 Problem Statement

Given a set of users $U = \{u_1, u_2, \ldots, u_i\}$, each associated with a set of *linked trajectories* over past time intervals, and a set of *unlinked trajectories* collected during a current time interval, the objective of the Trajectory-User Linking (TUL) task is to correctly associate each unlinked trajectory with its corresponding user in U.

Formally, let $Tr_u = \{Tr^{\tau}_{u_i} \mid u_i \in U, \tau \in T\}$ denote the set of all linked trajectories observed in earlier time intervals, and let $Tr^{\tau'} = \{Tr^{\tau'}_{?}\}$ represent the unlinked trajectories collected in the current time interval τ', where the generating users are unknown.

The goal is to learn a function

$$\mathcal{F} : Tr^{\tau'} \longrightarrow U$$

that maps each unlinked trajectory in $Tr^{\tau'}$ to the most likely user in U, using the information available from the historical (linked) trajectories Tr_u. This problem is inherently challenging due to the lack of explicit user identifiers, temporal shifts in mobility behavior, and potential noise in the data. Table 1.1 summarizes the preliminary notations and definitions.

TUL involves identifying which user most likely generated a given anonymized trajectory based on historical mobility behavior. The task is inherently complex due to the variability of human movement patterns, the sparsity or density of datasets, and differences in regional geography. Traditional models designed for TUL often rely on handcrafted architectures tailored to specific datasets, which limits their scalability and requires extensive retraining when applied to new domains [15].

TrajTRoPE, a recently developed foundation model for trajectory modeling, represents a shift from these task-specific approaches. It is pretrained on the Chengdu GPS dataset and subsequently fine-tuned on various downstream tasks including TUL, using the Xian dataset [19]. This pretraining-finetuning setup is designed to help the model learn generalized mobility representations across different urban contexts.

Table 1.1: Notations & Definitions [3]

Symbol	**Description**
$U = \{u_1, u_2, \ldots, u_i\}$	Set of users
t	Timestamp of a spatio-temporal point
ℓ	Geographical location (latitude, longitude)
$\langle t, \ell \rangle$	Spatio-temporal point
$Tr^{\tau}_{u_i}$	Trajectory of user u_i during interval τ
Tr_u	Set of all linked trajectories
$Tr^{\tau'}$	Set of unlinked trajectories in current interval τ'
M	Geographic map
$P = \{p_1, p_2, \ldots, p_{\lvert P \rvert}\}$	Set of Points of Interest (POIs)
$\mathcal{F}$	Mapping function from unlinked trajectories to users

1.4 Outcome of the project

TUL involves identifying which user most likely generated a given anonymized trajectory based on historical mobility behavior. The task is inherently complex due to the variability of human movement patterns, the sparsity or density of datasets, and differences in regional geography. Traditional models designed for TUL often rely on handcrafted architectures tailored to specific datasets, which limits their scalability and requires extensive retraining when applied to new domains.

TrajTRoPE, a recently developed transformer based for trajectory modeling, represents a shift from these task-specific approaches. What makes TrajTRoPE especially compelling is its potential for transferability the ability to be directly applied to novel datasets (such as GeoLife) for user classification and trajectory linkage. This aligns with the core promise of model: to significantly reduce the resource overhead associated with building new models for each task or dataset, while still achieving competitive or even superior performance.

2 Literature Review

In recent years, the domain of TUL predictions has seen significant advancements through various deep learning models. These models can be systematically categorized into three primary architectural themes: Recurrent Neural Networks (RNN) [24], Graph-based models, and Transformer-based model [26]. This review provides insights into some of the prominent TUL models, highlighting their features and contributions to the field.

2.1 Existing TUL Models

2.1.1 RNN-Based

RNNs have established themselves as a leading approach for modeling temporal sequences, particularly within the realm of TUL, which has evolved considerably. The seminal work, TULER (2017) [10], showcased the efficiency of RNNs in identifying users based on their motion patterns. TULER utilized the sequential characteristics of trajectory data through recurrent layers like Long Short-Term Memory (LSTM) [12] and Gated Recurrent Unit (GRU) [6]. The model begins by embedding check-in locations into a lower-dimensional space to capture semantic meaning, akin to techniques used in Natural Language Processing (NLP). These embeddings are subsequently processed by the RNN to uncover the fundamental motion patterns of users. TULER was designed as a semi-supervised learning model to effectively extract the complex features needed for distinguishing and classifying trajectories. Tests on real-world datasets revealed that TULER significantly outperformed traditional techniques such as Support Vector Machines (SVM) [8] and Linear Discrimant Analysis (LDA) [27], along with trajectory similarity measures like Longest Common Subsequence (LCSS) [25].

Nevertheless, TULER faced certain shortcomings. One notable issue is its performance on sparse trajectory data, where the model finds it challenging to link trajectories with a minimal number of check-ins. Additionally, TULER primarily concentrated on sequential patterns without fully addressing users' personalized movement preferences or the prior knowledge related to human mobility. It was also pointed out that TULER's approach of utilizing the entire dataset as a corpus for training location representations could lead to identity confusion when different trajectories share identical location sequences but with varying timestamps.

Building on TULER's foundation, TULVAE (2018) [32] merged RNNs with Variational Autoencoders (VAEs) [16] to tackle earlier limitations, particularly regarding data sparsity. By employing a semi-supervised learning framework, TULVAE utilizes extensive unlabeled data in conjunction with labeled datasets to enhance its performance. The VAE component facilitates TULVAE in learning the distributions of latent random variables, effectively modeling the uncertainty and noise inherent in trajectory data. This results in a more nuanced representation of user behaviors across time, capturing hierarchical and structural characteristics of trajectories with high-dimensional latent variables. Experi-

mental findings suggested that TULVAE often surpassed both TULER and other baseline models, highlighting the advantages of adopting a generative framework [32].

Despite its advancements, TULVAE is not without limitations. The latent representation it learns may not always prove effective, particularly with small or sparse datasets, potentially resulting from entangled representations. Moreover, TULVAE tends to incur greater time consumption compared to simpler RNN models like TULER, as it requires re-evaluating the generative likelihood for each class during training.

Following the developments of TULER and TULVAE, several additional RNN-based models have emerged to further enhance TUL performance:

- **DeepTUL (2020)** [21] introduced an attention mechanism to the recurrent network, enabling the model to learn the multi-periodic aspects of user mobility from historical trajectories while focusing on the most relevant sections of the trajectory for linking. DeepTUL also aimed to integrate multiple features influencing user mobility to model more intricate patterns and alleviate data sparsity by leveraging historical data.

- **AdattTUL (2021)** [9] presented a semi-supervised method that combines an attentional autoencoder with an adversarial network. This framework sought to model personalized movement preferences more effectively, improving the regularization of the latent space for trajectory classification and moving beyond the solely sequential patterns analyzed by earlier RNN approaches.

- **TGAN (2020)** [33] utilized a Generative Adversarial Network (GAN) [11] instead of a VAE to tackle data sparsity issues. TGAN's approach involves generating synthetic trajectories to augment training data for RNN models like TULER, potentially improving their performance. However, the challenges associated with training GANs and assessing the synthetic data's quality remain significant.

- **MainTUL (2023)** [4] introduced a mutual distillation learning network that integrates an RNN-based encoder with a temporal-aware Transformer encoder, enabling the model to capture both sequential patterns and long-range time dependencies. This design aims to maximize historical data utilization and incorporate rich contextual features such as categories of POI and temporal information, resulting in notable improvements, particularly with sparse data.

While TULER played a foundational role in applying RNNs to address the TUL challenge, TULVAE propelled the field forward by utilizing VAEs to manage data sparsity and model the distributions underlying user mobility. Subsequent research has built upon these foundations by integrating attention mechanisms, adversarial learning, and other advanced deep learning techniques, addressing persistent issues such as data sparsity, intricate mobility patterns, and the integration of diverse features for more accurate and insightful TUL. The trend points towards increasingly sophisticated architectures capable of better leveraging both labeled and unlabeled data, capturing richer contextual insights, and modeling the complexities of human movement.

2.1.2 Graph Based

Significant advancements in RNNs have impacted TUL; yet, the growing importance of Graph Neural Networks (GNNs) [34] for graph-structured data representation and processing offers a compelling alternative. The hypothetical model **AttnTUL (2023)** showcases how graph-based models can excel in TUL predictions by adeptly capturing the

complex relational dynamics among user interactions—something traditional sequential models often struggle with.

Recent research, including the **Multi-Scale Graph Attention Network** [18], has addressed key challenges within TUL modeling: missing context of the original trajectory, ignorance of spatial information, and high computational complexity [4]. This model features four essential components: check-in oriented graphs, node embeddings, trajectory embeddings, and user linking. The methodology involves segmenting geographical space into grids, with trajectory points mapped to respective grids [4]. Multi-scaled graphs are constructed by sampling trajectories at varying lengths, which are then processed through a multi-head self-attention-based GNN to derive node embeddings. These embeddings are subsequently refined using a bidirectional LSTM (bi-LSTM) [13], producing trajectory embeddings, and an Multilayer Perceptron (MLP) is employed for user linking. Experiments conducted on datasets such as Gowalla, Brightkite [5], and Foursquare [28] demonstrate promising results, underscoring the effectiveness of this model and the importance of each layer through an ablation study.

GNNs can effectively understand both direct and indirect relationships within user interaction networks, making them a strong candidate for use cases where user interactions form complex networks, such as on social media or e-commerce platforms [20]. This contrasts with traditional RNN-based approaches, which tend to process check-ins sequentially. Although models like DeepTUL and AdattTUL have integrated attention mechanisms to dynamically capture complex relationships among check-ins within a trajectory, and MainTUL has used a temporal-aware Transformer encoder to comprehend dependencies in enriched historical trajectories [4], they still operate within a fundamentally sequential paradigm.

The GNN framework envisioned by AttnTUL could represent user interactions and mobility patterns as graphs, with nodes signifying users or locations, and edges indicating interactions or transitions [3]. This representation could allow for a more straightforward capture of network effects and non-sequential dependencies. Moreover, the integration of GNNs could offer solutions to data sparsity—an enduring issue in TUL—by inherently aggregating information from neighboring nodes, thus providing a more effective means of harnessing sparse relational data.

The combination of GNNs and fully-connected layers in AttnTUL promotes multi-level analysis and captures detailed user behaviors, aligning with efforts in TUL to progress beyond basic sequential processing. In AdattTUL, the use of learnable spatio-temporal check-in embeddings processed through an attentional encoder signifies a move toward capturing more sophisticated user behavior by analyzing multiple preferences and the varying impact of check-ins [9]. Similarly, DeepTUL integrates various features while leveraging historical trajectories using an attention mechanism to capture diverse mobility patterns [21]. By integrating FC layers with a GNN, as proposed in AttnTUL, the model can achieve non-linear transformations and elegantly fuse features derived from the graph structure, potentially resulting in a more comprehensive understanding of user activity for personalized and context-aware recommendations [3].

Table 2.1 summarizes the key TUL models discussed, highlighting their methodologies and limitations across both RNN-based and Graph-based approaches.

Table 2.1: Summary of Various TUL Models

Classification	**Models**	**Methodology**	**Limitations**	**Reference**
RNN-Based	TULER (2017)	Uses LSTM/GRU to process embedded check-in locations, capturing sequential motion patterns in a semi-supervised framework.	Struggles with sparse data, overlooks personalized preferences, and may confuse identities with shared location sequences.	[10]
	TULVAE (2018)	Combines RNNs with VAEs to model latent variable distributions, handling data sparsity using semi-supervised learning.	Less effective with small/sparse datasets, higher training time due to generative likelihood re-evaluation.	[32]
	DeepTUL (2020)	Integrates attention mechanism to focus on relevant trajectory parts, leveraging historical data to model complex patterns.	Complexity increases with attention mechanism, may still face challenges with very sparse data.	[21]
	AdattTUL (2021)	Combines attentional autoencoder and adversarial network to model personalized preferences, improving latent space regularization.	Computationally intensive, effectiveness depends on quality of adversarial training.	[9]
Graph-Based	AttnTUL (2023)	Uses GNNs with multi-scale graph attention and bi-LSTM to capture relational dynamics and non-sequential dependencies in user interactions.	High computational complexity, requires careful tuning for sparse relational data.	[3]

3 Dataset Description & Preprocessing

3.1 Introduction

The dataset used in this project is the GeoLife GPS Trajectory Dataset, a richly annotated and widely used mobility dataset collected by Microsoft Research Asia over a span of more than five years, from April 2007 to August 2012 [31]. It includes trajectory data from 182 users, resulting in 18,670 trajectories that collectively span over 1.29 million kilometers and approximately 50,176 hours of movement [30]. Each trajectory is composed of timestamped points that include latitude, longitude, and altitude, and most are sampled at high frequency, with 91.5% of the data recorded every 1 to 5 seconds or 5 to 10 meters, offering a dense and fine-grained view of user movement.

What makes GeoLife particularly compelling is the diversity of its recorded activities. It captures both routine movements, such as commuting and shopping, and dynamic behaviors like hiking, sightseeing, and cycling. This makes it suitable for a wide range of spatial-temporal research areas, including mobility pattern mining, user activity recognition, privacy modeling, and location-based recommendation systems [29]. Although it contains data from over 30 cities worldwide, the majority of the trajectories were recorded in Beijing, providing a large, dense urban context ideal for spatial-temporal modeling and user behavior analysis.

This dataset has been extensively used in prior work on TUL, including by models such as TULER-L, TULER-G, Bi-TULER, TULVAE, DeepTUL, DPLink, T3S, GNNTUL, and most recently, AttnTUL [3]. These studies have established GeoLife as a benchmark dataset for evaluating user-level classification and trajectory modeling tasks.

In this project, the GeoLife dataset is leveraged to evaluate the performance of TrajTRoPE, a transformer based model.By benchmarking TrajTRoPE performance on GeoLife and comparing it against state-of-the-art task-specific models like AttnTUL, we aim to understand whether different architectural models can serve as robust, scalable alternatives to traditional TUL approaches.

3.2 Structure and Features of Dataset

We analyzed the spatial relationships between consecutive points within the trajectories as illustrated in Fig. 1. Our findings indicate that the majority of consecutive points are closely spaced, with approximately 70.1% of points being less than 50 meters. Additionally, 21.2% of points are between 50 and 100 meters apart, while smaller portions fall within greater distance intervals. This understanding of point distribution aids in assessing the granularity and resolution of the dataset, providing a foundation for evaluating user movement patterns.

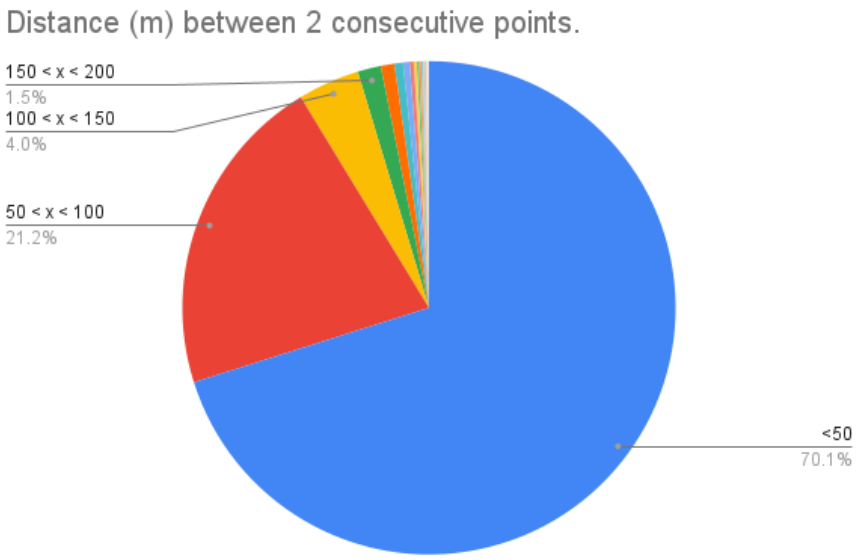


Figure 1: Distance (m) between 2 consecutive points

In terms of trajectory count per user, our analysis as seen in Fig. 2 demonstrates that 62.6% of users have 50 or fewer trajectories. A smaller portion, 14.8%, have between 50 and 100 trajectories, with the remainder distributed across larger trajectory counts. This distribution underscores the variability in user engagement and trajectory recording, offering insights into user behavior intensity and data availability.

Additionally, when examining the range of trajectory lengths, we observed a diverse set of trajectories. As seen in Fig. 3, many trajectories have lengths between 1000 to 2000 points and 2000 to 10000 points, highlighting varied complexity in user movements. This analysis further emphasizes the dataset's richness and the broad spectrum of recorded trajectories.

3.3 Preprocessing of the Data

3.3.1 Trip Preprocessing

To streamline the dataset for our analysis, we undertook a process of selective retention and modification of the columns. Specifically, we chose to retain the following columns: lat, long, date, and time. This selection focuses on the most pertinent information necessary for analyzing user trajectories and temporal patterns.

In addition to retaining existing columns, we introduced several new columns to enhance the analytical depth of the dataset:

1. **User Column**: We added a user column to identify individual users within the dataset. This facilitates the separation of trajectories based on specific participants, allowing for user-centric analysis.

2. **Trip Column**: To delineate distinct journeys made by users, we incorporated a trip column, where each individual trajectory is designated as a unique trip. This classification aids in organizing the dataset and assists in modeling user behavior during travel.

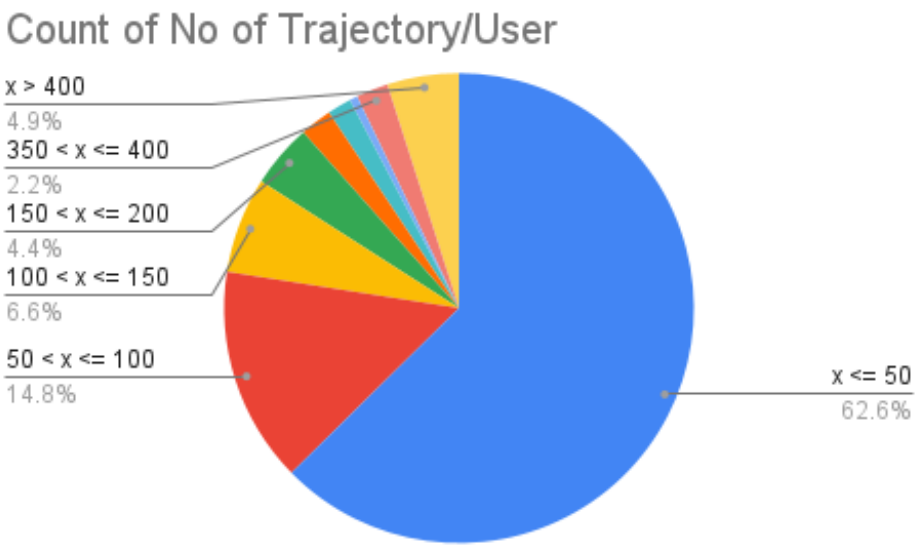


Figure 2: Trajectory Count per User

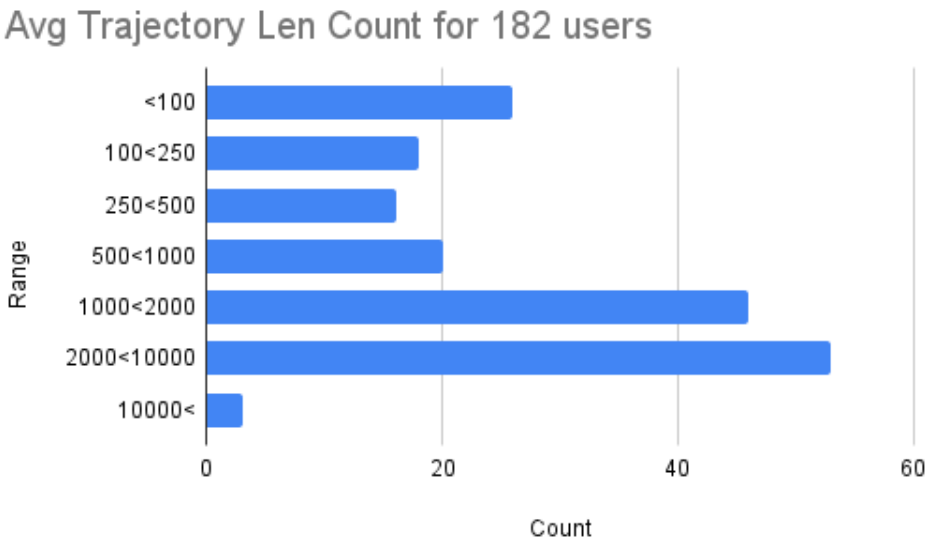


Figure 3: Average Trajectory Length Point Count for 182 users

3. **Sequence Index Column (seq_i)**: A seq_i column was introduced, which serves as a sequential index that starts from 0 and increments by 1 with each data point within a trajectory. Upon reaching the end of a trajectory, this counter resets to zero, effectively allowing for the identification of each point's position within its respective trajectory.

4. **Timestamp**: We added a timestamp column, representing the time of each recorded location in UNiplexed Information Computing System (Unix) time format. Unix time, also known as epoch time, counts the number of seconds since January 1, 1970 (excluding leap seconds) [23]. This format is advantageous for computational analysis, simplifying time calculations and comparisons.

5. **Delta Time Column (delta_t)**: Lastly, we created a delta_t column to quantify the temporal progression from the first point of each trajectory to every subsequent point. This column records the elapsed time in seconds since the initiation of the trajectory, providing insights into the duration of trips and the timing of movements along each trajectory.

Table 3.1 mentions the dataset columns retained and introduced during trip preprocessing.

3.3.2 POI Preprocessing

To support the analysis of user mobility patterns in the GeoLife dataset, we focused on associating trajectory points with relevant POIs. Due to limitations in the number of API calls available, it was not feasible to query POIs for every individual trajectory point. As a result, we sampled every 10th point from each user's trajectory to reduce the number of API requests while still preserving the overall structure and trends of user movement. In addition, a search radius of 250 meters was applied around each sampled point to retrieve nearby POIs. This distance was chosen based on the observation that over 90 percent of consecutive trajectory points in GeoLife were spaced less than 200 meters apart [31], making 250 meters a practical threshold for capturing relevant surrounding landmarks without unnecessary redundancy.

Once the POIs were collected, we organized and filtered them for analysis. The dataset contained separate CSV files for each of the 182 users, with each file potentially containing

Table 3.1: Finalized Dataset Columns after Trip Preprocessing

Type	**Column Name**	**Description**
Retained	**lat**	Latitude of the recorded location.
	long	Longitude of the recorded location.
	date	Date of the recorded location.
	time	Time of the recorded location.
Introduced	**user**	Identifies individual users in the dataset.
	trip	Designates each trajectory as a unique trip.
	seq_i	Sequential index starting at 0, incrementing per point in a trajectory, resetting at the end.
	timestamp	Time of each location in Unix time format (seconds since Jan 1, 1970).
	delta_t	Elapsed time in seconds from the start of the trajectory to each point.

up to three POIs per sampled point. A script was written in python [**?**] to iterate through all user files, check if each file existed, and then extract key POI information from the valid ones. For each record, the script retrieved the POI name, type, latitude, longitude, distance from the user's position (in meters), and direction (in degrees). To ensure data quality, POI entries with missing names or those that appeared to be placeholder numeric values were excluded. All valid POIs were then compiled into a single dataset.

To avoid duplication, only the first occurrence of each unique POI name was retained. This de-duplication process helped streamline the dataset and ensured that repeated entries across users or trajectories did not bias subsequent analysis. The final output, a clean and unified dataset of unique POIs, was saved as unique_poi_all.csv. This dataset now serves as a foundational component for analyzing how users interact with different urban spaces and for performing tasks such as TUL, mobility pattern discovery, and behavioral modeling in a spatial context.

4 Methodology

In this section, we present a detailed description of the TrajTRoPE framework for user identification from trajectory data. Specifically, we modify the original TrajTRoPE model to adapt it for a user labeling (TUL) task. In contrast to the original design, we remove the masking and recovery schemes from the original model, thereby focusing entirely on capturing rich representations based on the spatial, temporal, and POI (Point of Interest) modalities. As demonstrated in Fig. 4, our complete pipeline is composed of three primary components: the STRFormer (transformer) encoder, the STRPE encoder, and the TUL classification head.

4.1 Overview

The original framework was designed to handle a wide variety of trajectory inference tasks. It achieves this by encoding each trajectory point with independent embeddings corresponding to space, time, and POI information. In our adaptation of the model, we deliberately remove the trajectory masking and recovery mechanisms in order to streamline the design specifically for the user identification task.

Each point within a trajectory sequence is organized according to the following structure:

$$p_i = (\mathrm{lng}_i, \mathrm{lat}_i, t_i, l_i),$$

where lng_i and lat_i refer to the longitude and latitude coordinates respectively, t_i denotes the timestamp associated with the point, and l_i indicates the corresponding POI annotation.

4.2 Transformer

The STRFormer module is responsible for transforming each trajectory point into embeddings across the three defined modalities.

4.2.1 Trajectory Modalities and Embeddings

Spatial modality

We utilize the Universal Transverse Mercator (UTM) coordinate projection system and apply a normalization step with respect to the regional center coordinates:

$$x_i = \frac{\mathrm{UTM}(\mathrm{lng}_i) - \mathrm{UTM}(\mathrm{lng}^{\mathrm{cen}})}{s_x}, \quad y_i = \frac{\mathrm{UTM}(\mathrm{lat}_i) - \mathrm{UTM}(\mathrm{lat}^{\mathrm{cen}})}{s_y}, \tag{4.1}$$

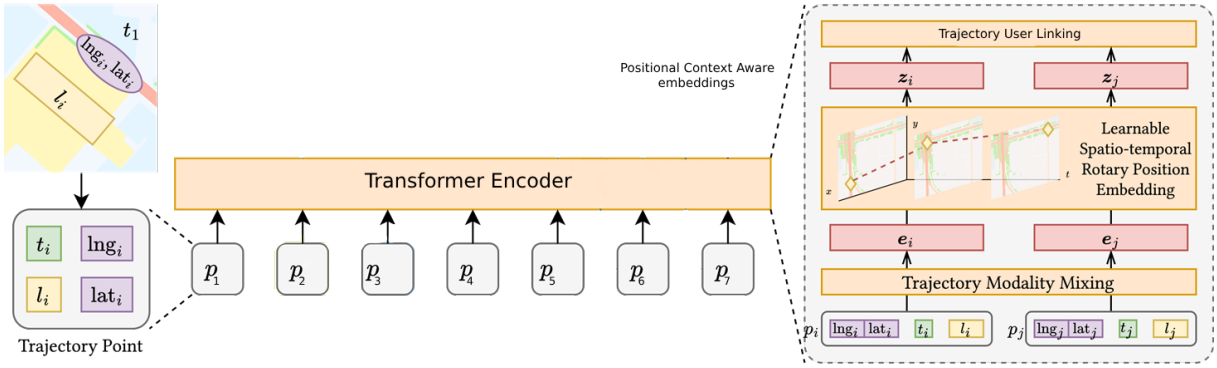


Figure 4: Diagram of the pipeline used in this work

with scaling factors set to $s_x = s_y = 4000$. The normalized spatial features are then passed through a linear layer to obtain the spatial embedding vector $e_i^x \in R^d$ as follows:

$$e_i^x = \text{Linear}((x_i, y_i)).$$

Temporal modality

Each timestamp t_i is first converted into four periodic features:

$$\text{Time Features} = [\text{DayOfWeek}(t_i), \text{Hour}(t_i), \text{Minute}(t_i), \text{TimeGap}(t_i)].$$

These periodic features are encoded using learnable Fourier embeddings for better periodic pattern representation, and then concatenated and fed into a linear transformation:

$$e_i^t = \text{Linear}(\text{Concat}(\text{Fourier}(t_i))) \in R^d.$$

POI modality

For the POI information, each textual POI descriptor desc_i is first embedded using a text embedding model, and then subsequently transformed through a linear layer to match the feature dimension:

$$e_i^l = \text{Linear}(\text{Embedding}_{\text{text}}(\text{desc}_i)) \in R^d.$$

4.2.2 Trajectory Modality Mixing

The resulting modality-wise embeddings for each point (e_i^x, e_i^t, e_i^l) are then concatenated and processed via a one-layer Transformer encoder (`TransEnc`). Afterwards, a mean pooling operation is applied to integrate the modalities into the final point-wise embedding:

$$e_i = \text{MeanPool}(\text{TransEnc}([e_i^x, e_i^t, e_i^l])), \tag{4.2}$$

effectively capturing inter-modal relationships.

4.3 STRPE: Spatio-Temporal Rotary Position Embedding

To explicitly model the relative spatial structure between trajectory points, we apply a spatio-temporal rotary position embedding (STRPE) scheme [2]. Given two points with

coordinates (x_i, y_i) and (x_j, y_j), we construct the rotary embedding matrix as:

$$\mathbf{R}_\theta(x,y) = \begin{bmatrix} \cos(\theta_1 x), & \cos(\theta_2 x), & \ldots, & \cos(\theta_{d/2} x), & \cos(\theta_1 y), & \cos(\theta_2 y), & \ldots, & \cos(\theta_{d/2} y) \\ \sin(\theta_1 x), & \sin(\theta_2 x), & \ldots, & \sin(\theta_{d/2} x), & \sin(\theta_1 y), & \sin(\theta_2 y), & \ldots, & \sin(\theta_{d/2} y) \end{bmatrix}. \tag{4.3}$$

Following this rotary embedding, the input queries to the attention mechanism are designed as:

$$\Phi(x,y) = W_q \mathbf{R}_\theta(x,y),$$

where $W_q \in R^{2d \times d}$ is a learnable projection matrix.

Subsequently, for each input embedding e_i, the queries (q_i), keys (k_j), and values (v_j) are computed:

$$q_i = \Phi(x_i, y_i) W_q e_i, \tag{4.4}$$
$$k_j = \Phi(x_j, y_j) W_k e_j, \tag{4.5}$$
$$v_j = W_v e_j, \tag{4.6}$$

with $W_k, W_v \in R^{d \times d}$ as additional learnable matrices.

The attention-based output embedding h_i is then obtained by:

$$h_i = \sum_{j=1}^{n} \text{softmax}\left(\frac{q_i^\top k_j}{\sqrt{d}}\right) v_j. \tag{4.7}$$

4.4 Trajectory User Linking (TUL)

4.4.1 Input Representation

Following the generation of modality-specific embeddings, we prepare the trajectory for the user labeling stage. Each point in the trajectory is represented by its spatial, temporal, and POI embeddings, which have been projected separately into vectors of dimension R^d. To handle the sequential nature of the data and provide explicit boundary information for the model, we introduce two special tokens: a start token $[s]$ at the beginning and an end token $[e]$ at the conclusion of each trajectory. These special tokens help the model delineate complete trajectory representations during encoding, ensuring better initialization and closure signals for sequence-level understanding.

4.4.2 Encoding and Pooling

Once the tokenized trajectory, including the $[s]$ and $[e]$ markers, has been prepared, the sequence is processed through the STRPE-enhanced attention mechanism. This yields a set of contextualized embeddings $\{h_1, h_2, \ldots, h_T\}$, where each h_i captures information aggregated from other points in the sequence according to their spatial and temporal relationships.

To obtain a single, condensed representation of the entire trajectory, we apply a mean pooling operation over the sequence. Specifically, the embeddings of all points, including the start and end tokens, are averaged to produce a trajectory-level embedding denoted as $\bar{h}$. This summarized vector $\bar{h}$ effectively encodes the holistic characteristics of the user's trajectory in a fixed-dimensional latent space, providing a compact input for the final classification stage.

4.4.3 User Prediction Head

The pooled trajectory representation $\bar{h}$ is then fed into a lightweight prediction head, consisting of a linear transformation followed by a softmax activation function. The linear transformation uses a learnable matrix W of shape $U \times d$, where U represents the total number of user classes, and d is the embedding size. A bias term $b \in R^U$ is also added.

The output is a probability distribution over the U possible users, denoted by $\hat{y}$, where each element $\hat{y}_u$ corresponds to the predicted likelihood that the input trajectory belongs to user u. This step enables the model to effectively map a pooled trajectory embedding into a discrete prediction space representing the user identities.

4.4.4 Training Objective

For model training, we employ a standard supervised learning approach using the cross-entropy loss function. During each training iteration, the predicted distribution $\hat{y}$ is compared against the ground-truth user label, which is represented as a one-hot encoded vector y.

The cross-entropy loss penalizes deviations between the predicted and true distributions by summing over the negative log-probabilities of the correct classes. By minimizing this loss across the training dataset, the model progressively improves its ability to correctly associate trajectory patterns with corresponding users, optimizing the parameters of both the encoder components and the prediction head in an end-to-end manner.

5 Experimental Design

5.1 Experimental Settings

To evaluate the performance of the proposed model on the trajectory-user linking task, we conducted experiments using selected user subsets from the GeoLife GPS trajectory dataset. Two different experimental configurations were designed to examine the behavior of the model under varying trajectory lengths and user data.

As seen in Table 5.1, data from **89 users** was used for the first configuration. A total of 8317 trajectories were selected, each truncated to a fixed maximum length of 1000 spatio-temporal points. The model was trained with a batch size of 32 for 30 epochs using a learning rate of 0.0001. The architecture included two layers of rotary positional encoding (RoPE) to capture temporal dependencies across the sequence.

In the second configuration, we used data from **56 users**. In this setup, 7602 trajectories were selected, and truncated to a maximum length of 1000. Rest of the training parameters include a batch size of 32, 60 training epochs, a learning rate of 0.0002, and two RoPE layers.

5.2 Evaluation Metrics

To assess the performance of the model on the trajectory-user linking task, we employed a set of well-established evaluation metrics commonly used in multi-class classification problems. These metrics were selected to provide a comprehensive understanding of the model's accuracy, robustness, and ability to generalize across users. The evaluation includes both top-k accuracy measures and macro-averaged classification metrics, ensuring

Table 5.1: Experimental Settings for Trajectory-User Linking

Parameter	**Setup 1 (89 users)**	**Setup 2 (56 users)**
Number of Trajectories	8317	7602
Max Trajectory Length	1000	1000
Batch Size	32	32
Epochs	30	60
Learning Rate	0.0001	0.0002
Embedding dimension	64	64
d_model	128	128
RoPE Layers	2	2

that performance is not only judged by overall correctness but also by the balance across all user classes. The following sections describe each metric in detail and explain how it contributes to the overall evaluation framework.

5.2.1 Accuracy at 1 ($ACC@1$)

Measures the percentage of predictions where the model's top predicted class exactly matches the true class [17]. It is widely used in classification and ranking tasks where a single correct answer is expected. For instance, in deep learning benchmarks such as ImageNet, a model is said to be correct only if the right label is at the top of its predictions.

$$\text{Acc@1} = \frac{1}{N}\sum_{i=1}^{N} 1\left(y_i \hat{y}_i^{(1)}\right) \tag{5.1}$$

In the context of trajectory-user linking, Acc@1 evaluates how often the model correctly identifies the true user as the first choice, making it a strict and precise indicator of model performance.

5.2.2 Accuracy at 5 ($ACC@5$)

Measures the percentage of predictions where the true class appears among the top five predicted classes [17]. This metric is useful in tasks where multiple outcomes could be plausible or where there is ambiguity in the labeling. For example, even if the correct user is not the model's first choice, placing it within the top five predictions still shows that the model has learned useful information.

$$\text{Acc@5} = \frac{1}{N}\sum_{i=1}^{N} 1\left(y_i \in \{\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \ldots, \hat{y}_i^{(5)}\}\right) \tag{5.2}$$

Acc@5 is commonly used in large-scale classification tasks such as image recognition and is a relaxed measure that allows us to understand how well the model narrows down the correct prediction.

5.2.3 Macro Precision ($MacroP$)

Macro precision is the average of the precision scores computed independently for each class [14]. Precision is defined as the ratio of true positives to the sum of true positives and false positives. Useful in multi-class classification tasks where balanced performance across all classes is required, especially in the presence of class imbalance.

$$\text{Macro Precision} = \frac{1}{C}\sum_{j=1}^{C} \frac{TP_j}{TP_j + FP_j} \tag{5.3}$$

Macro precision evaluates how well the model avoids false positives across all user classes, giving equal importance to each class. This is helpful when it is important to reduce the rate of incorrect user associations uniformly.

5.2.4 Macro Recall ($MacroR$)

Macro recall is the average of the recall scores in all classes, where recall is the ratio of true positives to the sum of true positives and false negatives [1]. Common in imbalanced

classification problems like medical diagnosis or NLP, where missing a minority class prediction is critical.

$$\text{Macro Recall} = \frac{1}{C}\sum_{j=1}^{C}\frac{TP_j}{TP_j + FN_j} \tag{5.4}$$

Macro recall measures how well the model identifies all relevant instances of each class, giving equal importance to all classes regardless of how frequently they appear. It reflects the model's sensitivity across users.

5.2.5 Macro F1 Score ($MacroF1$)

Macro F1 is the average of the F1 scores across all classes [22]. F1 score is the harmonic mean of precision and recall for each class. Widely used in multi-class and imbalanced classification tasks to get a single, balanced measure of model performance.

$$\text{Macro F1} = \frac{1}{C}\sum_{j=1}^{C}\frac{2 \cdot \text{Precision}_j \cdot \text{Recall}_j}{\text{Precision}_j + \text{Recall}_j} \tag{5.5}$$

In trajectory-user linking, where incorrect identification or failure to identify a user are both problematic, macro F1 provides a comprehensive view of model performance across the entire user base.

6 Results and Analysis

6.1 Performance Comparison

In this section, a comparative analysis of the model's performance under two different experimental configurations involving 89 users and 56 users is presented. The configurations were designed with slight variations to explore the effect of hyperparameters and training dynamics on the TUL task.

6.1.1 Setting 1: User Count 89 (Trajectory Length $\leq$ 1000)

The first setup focused on 89 users, where trajectories were constrained to have no more than 1000 points. The model was trained with a batch size of 32, an embedding size of 64, and a learning rate of 0.0001, over 30 epochs, utilizing two layers of rotary positional encoding (RoPE). The dataset contained a total of 5,941,128 trajectory points distributed across 8,317 trajectories. Under this configuration, the model achieved an Accuracy@1 of 68.01% and an Accuracy@5 of 87.43%. The macro-averaged metrics, which provide a balanced evaluation across all user classes, yielded a Macro Precision of 50.65%, a Macro Recall of 53.08%, and a Macro F1 score of 50.63%. These results reflect the model's strong capability to predict the correct user with high top-1 accuracy, while also maintaining balanced sensitivity and specificity across different users.

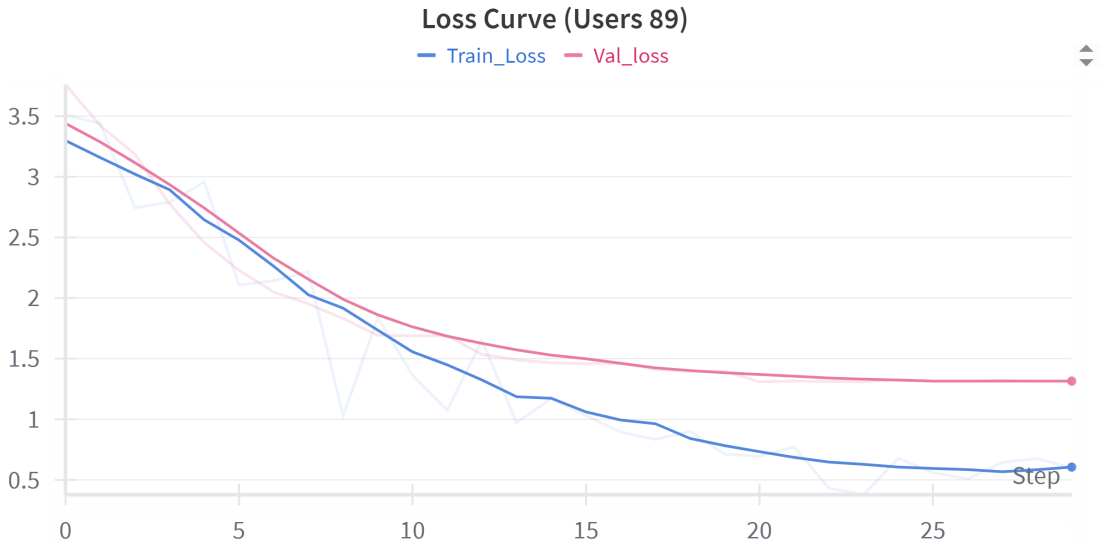
6.1.2 Setting 2: User Count 56 (Trajectory Length $\leq$ 1000)

The second setup involved 56 users, where the model was trained under slightly tweaked hyperparameters to accommodate the more challenging nature of the data. Specifically, the number of epochs was increased to 70 and the learning rate was adjusted to 0.0002, while maintaining the batch size and embedding size identical to the first setup. The dataset for 56 users comprised 5,450,141 trajectory points across 7,602 trajectories. Despite these adjustments, the performance exhibited some limitations compared to the previous setting (89 users). The model achieved an Accuracy@1 of 62.76% and an Accuracy@5 of 87.5%. The macro-averaged metrics stood at a Macro Precision of 45.19%, a Macro Recall of 46.48%, and a Macro F1 score of 44.51%. Although the ACC@5 remained comparable to the first setup, the lower macro scores suggest that the model faced greater difficulty in consistently identifying users across all classes under this configuration.

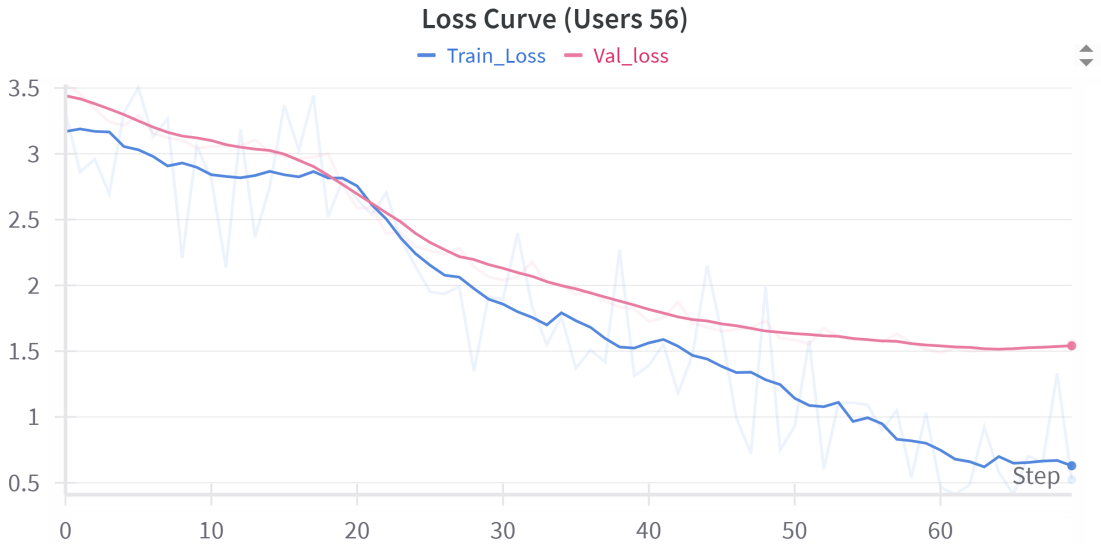
6.1.3 Analysis and Insights

A direct comparison between the two setups as shown in table **??** reveals that while both achieved strong top-5 prediction capabilities, the model performed more reliably for User Count 89, especially in terms of Macro Precision, Macro Recall, and Macro F1. Fig. 5-a demonstrates that the model converges much faster for User Count 89 setting (within

30 epoch) in comparison to the User Count 56 setting (within 70 epoch), as seen in Fig. 5-b. The longer training schedule and higher learning rate for User 56 helped mitigate some issues but could not entirely bridge the performance gap. These results suggest that beyond hyperparameter optimization, intrinsic differences in user mobility patterns and trajectory complexity significantly influence model performance. Thus, the richness and consistency of user behavior captured in the trajectories play a vital role in facilitating accurate and robust user identification.



(a) Loss Curve for User 89 Setting



(b) Loss Curve for User 56 Setting

Figure 5: Train and Validation Loss Curves.

Table 6.1: Performance Comparison Between TrajTRoPE and AttnTUL for Users 89 and 56

Metric	User 89		User 56	
	TrajTRoPE	**AttnTUL**	**TrajTRoPE**	**AttnTUL**
ACC@1 (%)	**68.33**	53.92	**63.6**	61.37
ACC@5 (%)	**88.58**	79.47	**86.21**	85.87
Macro-P (%)	**51.55**	49.94	48.47	**59.20**
Macro-R (%)	**53.41**	50.29	49.36	**59.29**
Macro-F1 (%)	**51.34**	48.24	47.57	**58.53**

6.2 Ablation Study

To better understand the contribution of each architectural component to the overall performance of the model, we conducted an ablation study. Specifically, we compared the full model incorporating both the Transformer and the RoPE Encoder against a simplified variant utilizing only the Transformer. All experiments were conducted on the subset of User 89 setting, with trajectories constrained to a maximum length of 1000 points.

6.2.1 Full Model (Transformer with RoPE Encoder)

In the full model configuration, the Transformer was complemented by two layers of RoPE encoder to enhance the capturing of sequential dependencies. The training configuration utilized a batch size of 32, an embedding size of 64, a learning rate of 0.0001, and 30 training epochs. The dataset included a total of 5,941,128 points distributed across 8,317 trajectories.

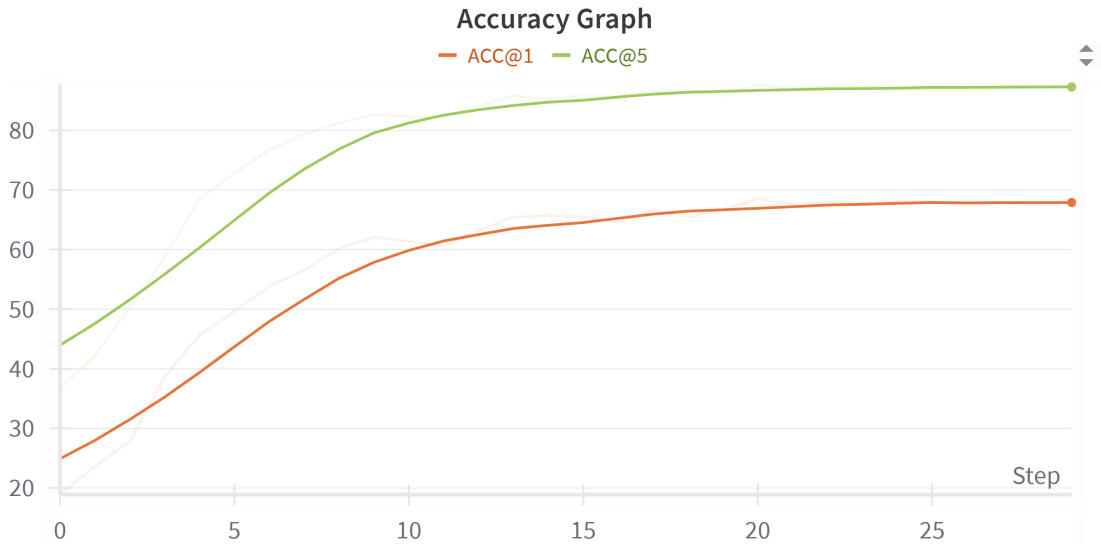
Under this setup, the model achieved an Accuracy@1 of 68.01% and an Accuracy@5 of 87.43%, as seen in Fig. 6-a. The macro-averaged metrics were also strong as demonstrated in Fig. 6-b, with a Macro Precision of 50.65%, a Macro Recall of 53.08%, and a Macro F1 score of 50.63%. These results demonstrate that the full model configuration provides a balanced and robust framework for capturing both spatial and temporal information necessary for effective trajectory-user linking.

6.2.2 Transformer Only (without RoPE Encoder)

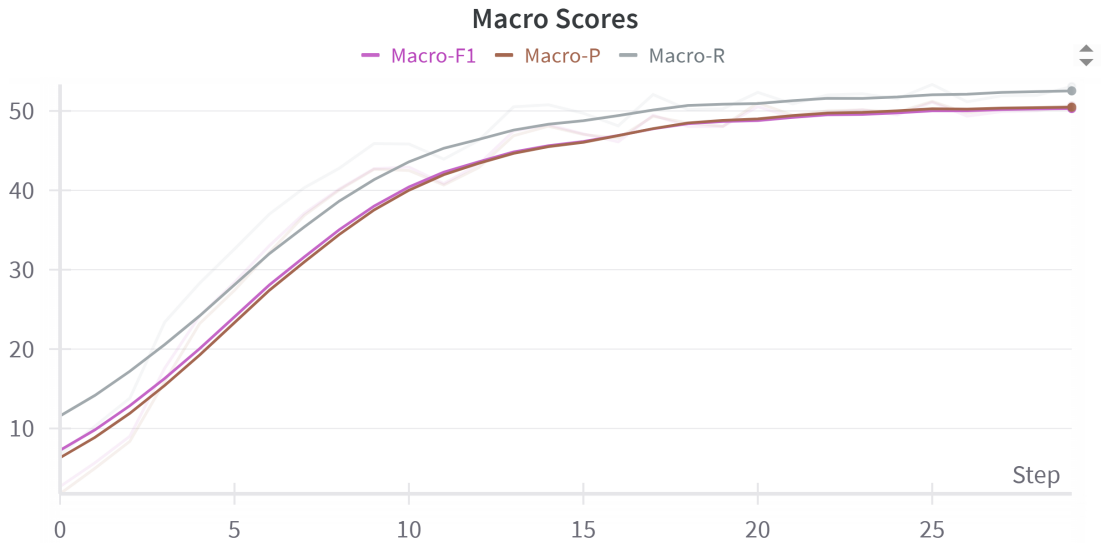
To isolate the impact of the RoPE Encoder, we conducted a second experiment where the model architecture included only the Transformer component, removing the RoPE layers. The training settings remained largely consistent, except that the number of training epochs was increased to 70 which will be explained in the later section.

In this setting, the model achieved an ACC@1 of 65.43% and an ACC@5 of 86.77%, as seen in Fig. 7-a. In comparison to the full model, the macro-averaged metrics exhibited a slight decline as demonstrated in Fig. 7-b, with a Macro Precision of 47.48%, a Macro Recall of 47.56%, and a Macro F1 score of 47.56%.

Although the Transformer-only configuration still produced competitive results, the performance drop relative to the full model suggests that the RoPE Encoder plays a crucial role in enhancing the model's ability to learn temporal patterns within trajectories. The degradation observed across all evaluation metrics underscores the importance



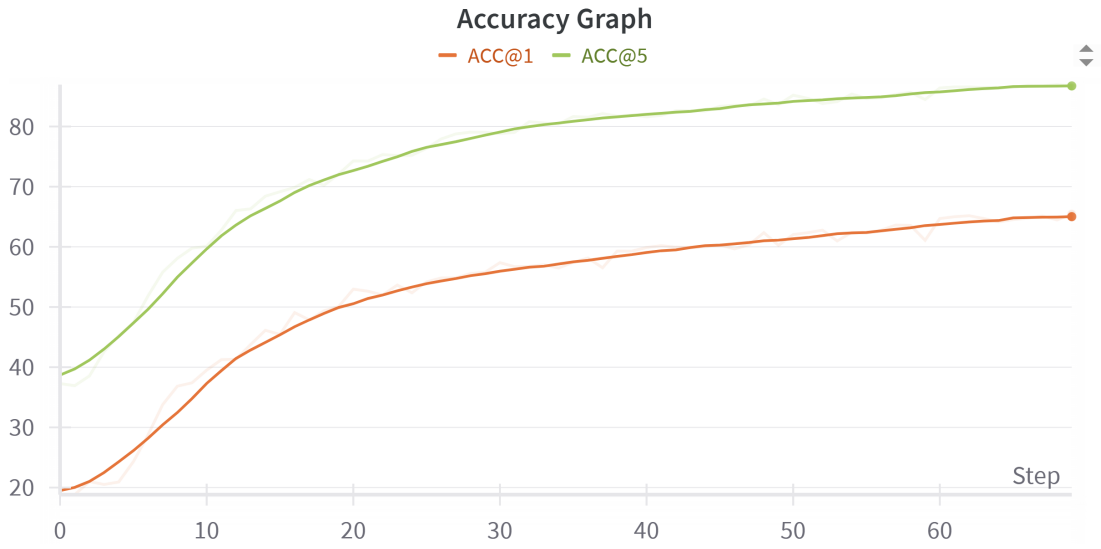
(a) Accuracy Graph



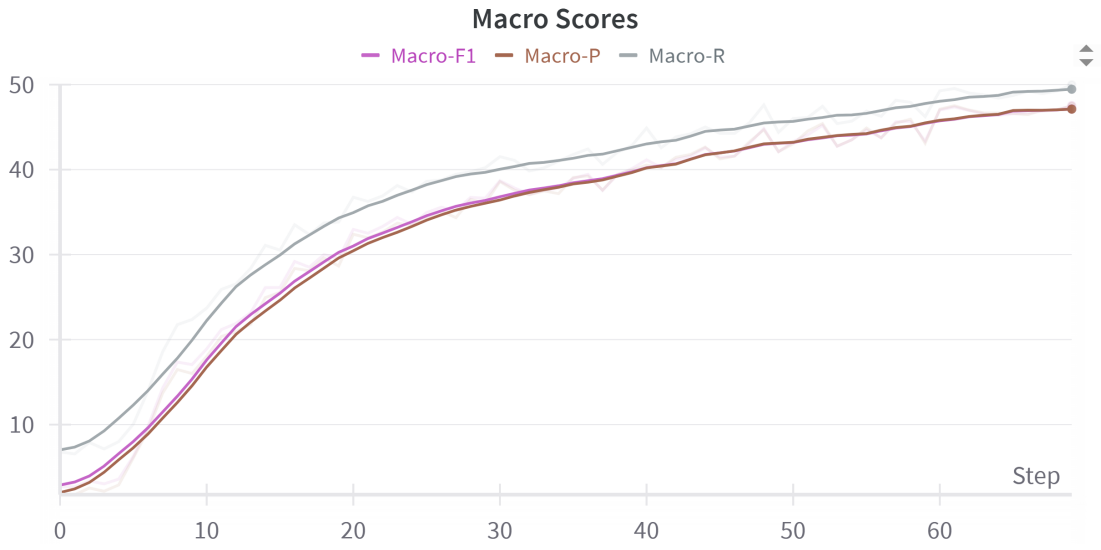
(b) Macro Scores

Figure 6: Performance Metrics Over Training Steps for Full Model: (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score.

of incorporating explicit positional information when modeling spatio-temporal mobility data.



(a) Accuracy Graph



(b) Macro Scores

Figure 7: Performance Metrics Over Training Steps for Transformer-only Variant: (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score.

6.2.3 Importance of RoPE Encoder

The comparison between the full model and the Transformer-only variant reveals that the addition of the RoPE Encoder contributes substantially to overall performance. In particular, improvements in both top-1 accuracy and macro-averaged scores indicate that positional encoding enhances not only the model's confident predictions but also its ability to generalize across diverse user behaviors. Moreover, it is worth noting that in the Transformer-only setting, the model required 70 epochs — an additional 40 epochs beyond the original 30 — to achieve acceptable convergence. This extension was necessary because the training loss and validation loss exhibited significantly slower convergence when the positional encoding was removed. Consequently, the Transformer-only configuration not only resulted in lower performance but also demanded substantially more training time and computational resources. These findings further validate the architectural design choice of integrating rotary positional encoding into the Transformer backbone, demonstrating that it improves both the effectiveness and the efficiency of the model for trajectory-user linking tasks.

6.3 Feature Importance

To rigorously assess the contribution of auxiliary semantic features to the trajectory-user linking task, we conducted a feature importance study focused on the inclusion of POI information. Two experimental setups were designed: one incorporating POI features into the input representations and another excluding them. Both setups were evaluated on the same dataset, ensuring that any observed differences could be attributed directly to the presence or absence of POI information.

6.3.1 Model Performance with POI Features

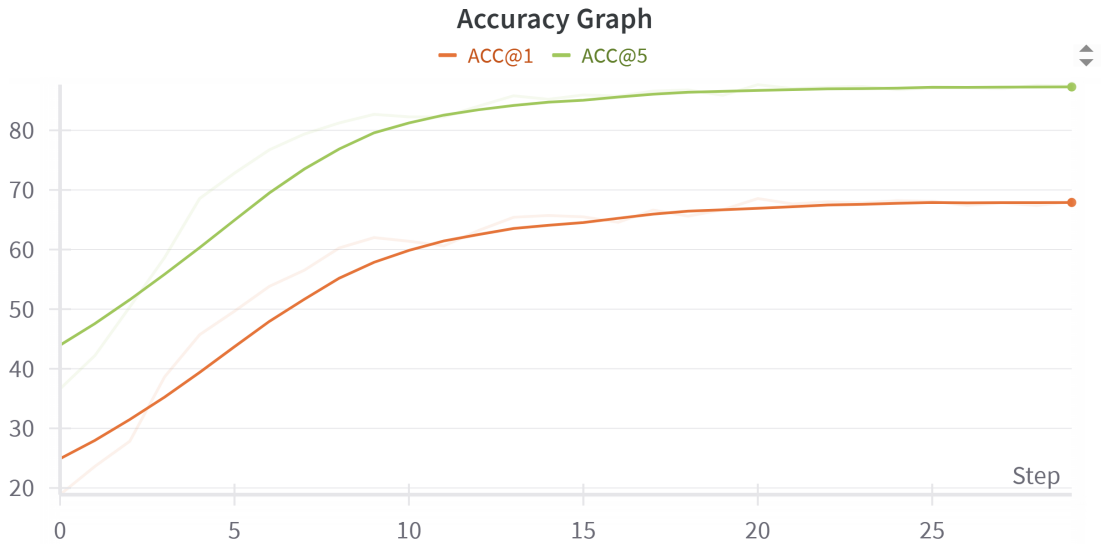
In the first configuration, the model was trained using trajectory data enriched with POI information. The training was conducted with a batch size of 32, an embedding size of 64, a learning rate of 0.0001, and two RoPE layers over 30 epochs. The dataset consisted of 5,941,128 points distributed across 8,317 trajectories.

The model demonstrated strong performance, achieving an Accuracy@1 of 68.01% and an Accuracy@5 of 87.43%, as seen in Fig. 8-a. The macro-averaged metrics were similarly robust as demonstrated in Fig. 8-b, with a Macro Precision of 50.65%, a Macro Recall of 53.08%, and a Macro F1 Score of 50.63%. These results underscore the model's capacity to effectively leverage contextual POI information to enhance user identification performance across a wide range of classes.

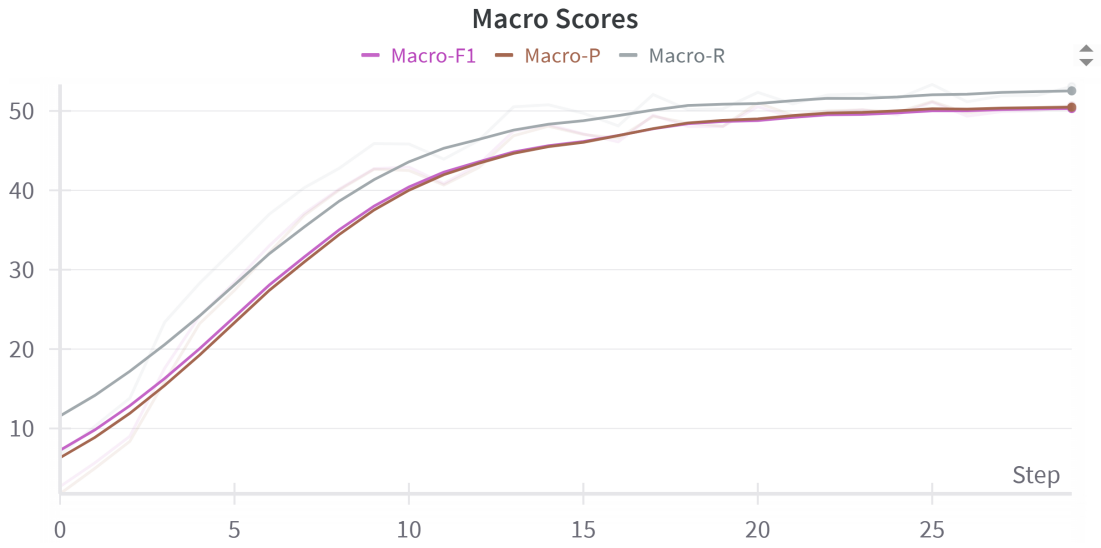
6.3.2 Model Performance without POI Features

In the second configuration, the model architecture and all hyperparameters remained unchanged; however, POI features were deliberately omitted from the input. Despite operating under otherwise identical conditions, the model's performance deteriorated dramatically.

The Accuracy@1 fell to 17.74%, while Accuracy@5 dropped to 42.28%, as seen in Fig. 9-a. More critically, the macro-averaged scores collapsed as demonstrated in Fig. 9-b, with Macro Precision, Macro Recall, and Macro F1 Score dropping to 4.47%, 7.25%, and 4.94% respectively. This significant degradation highlights the model's inability to



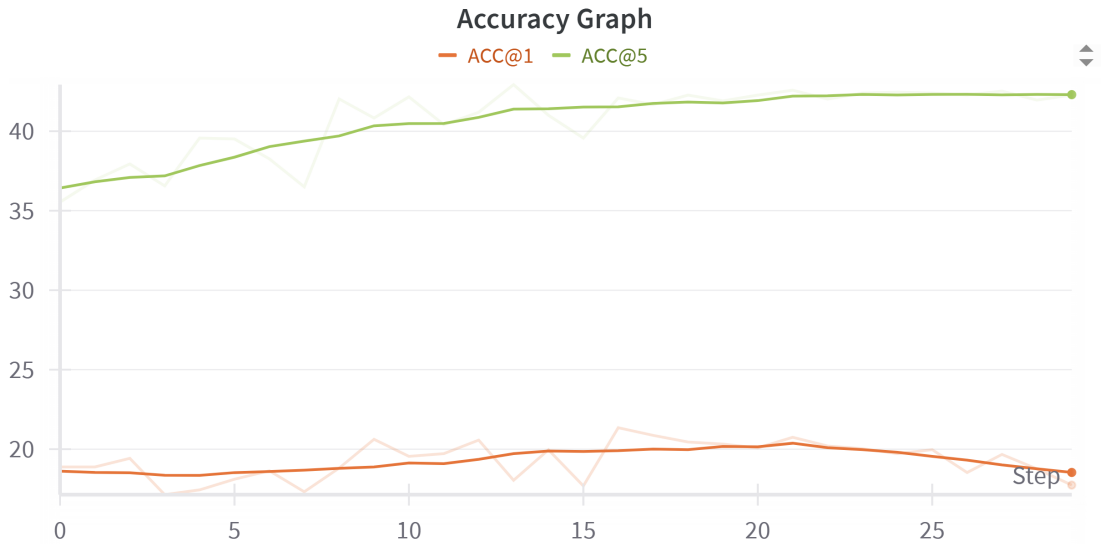
(a) Accuracy Graph



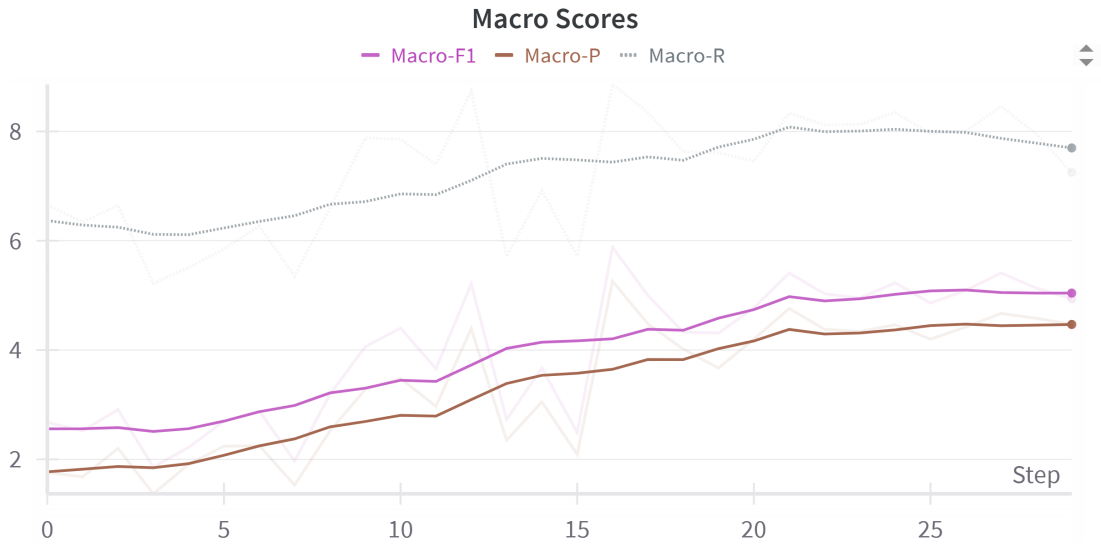
(b) Macro Scores

Figure 8: Performance Metrics With POI : (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score.

meaningfully distinguish between user trajectories based solely on raw spatio-temporal signals.



(a) Accuracy Graph



(b) Macro Scores

Figure 9: Performance Metrics Without POI : (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score.

6.3.3 Importance of POI

The comparison between these two settings provides compelling evidence of the pivotal role played by POI information in trajectory-user linking. While temporal and spatial sequences encode important behavioral patterns, they often lack sufficient discriminative power when considered in isolation. The inclusion of POI features substantially enriches

the semantic representation of user mobility, enabling the model to capture user-specific preferences and visitation behaviors that are otherwise imperceptible.

The stark decline in performance when POIs are excluded indicates that contextual information about visited locations serves as an indispensable signal for disambiguating user identities. These findings reinforce the broader notion that successful modeling of human mobility requires not only structural modeling of trajectories but also the incorporation of high-level semantic features that ground those trajectories in meaningful real-world contexts.

6.4 Impact of Tokenization on Model Performance

To further investigate the architectural components influencing the effectiveness of our model, we conducted an experiment focused on the role of tokenization. In our context, tokens refer to the structured encoding of various trajectory attributes—such as POIs, temporal cues, and categorical features—into fixed-length symbolic representations that are passed to the Transformer. The hypothesis is that these tokens provide additional structural guidance to the model, enhancing its ability to distinguish between different user behaviors.

6.4.1 Model Performance with Tokens

In the first configuration, tokenization was enabled, allowing the model to process structured representations of input features alongside raw trajectory data. The model was trained on User 89 with a maximum trajectory length of 1000 points, using a batch size of 32, an embedding size of 64, and a learning rate of 0.0001 for 30 epochs with two RoPE layers. The dataset comprised 5,941,128 total points across 8,317 trajectories.

The performance of the model in this setting was strong across all metrics. It achieved an Accuracy@1 of 68.01%, Accuracy@5 of 87.43% as seen in 10-a, Macro Precision of 50.65%, Macro Recall of 53.08%, and a Macro F1 Score of 50.63%, as demonstrated in 10-b. These results confirm that the tokenized structure significantly contributed to both high-confidence predictions and balanced class-level performance.

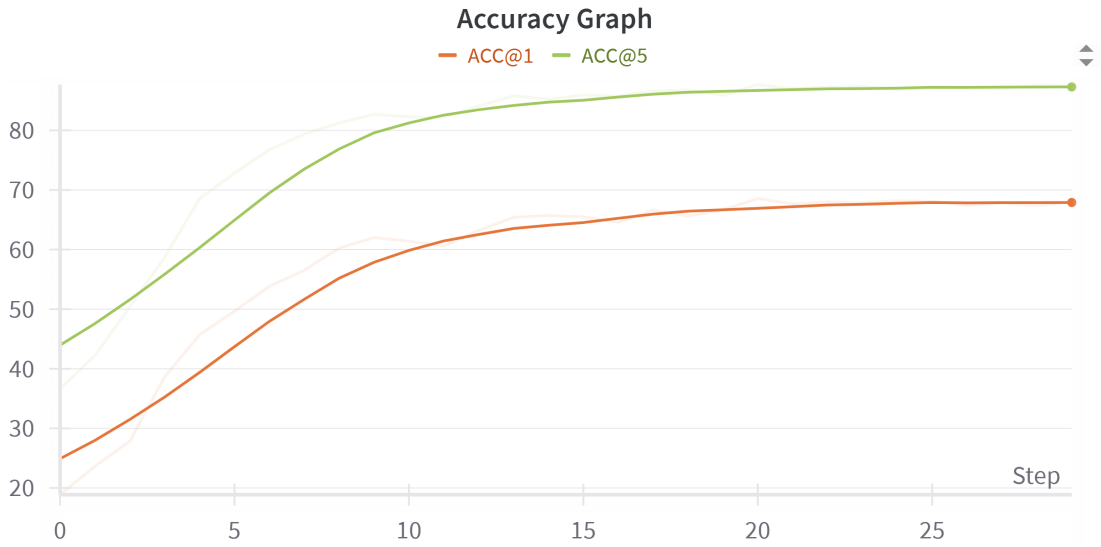
6.4.2 Model Performance without Tokens

In the second configuration, tokenization was disabled, meaning the model received only raw numerical values without any symbolic abstraction of trajectory features. All other hyperparameters and dataset conditions remained identical to the previous setup.

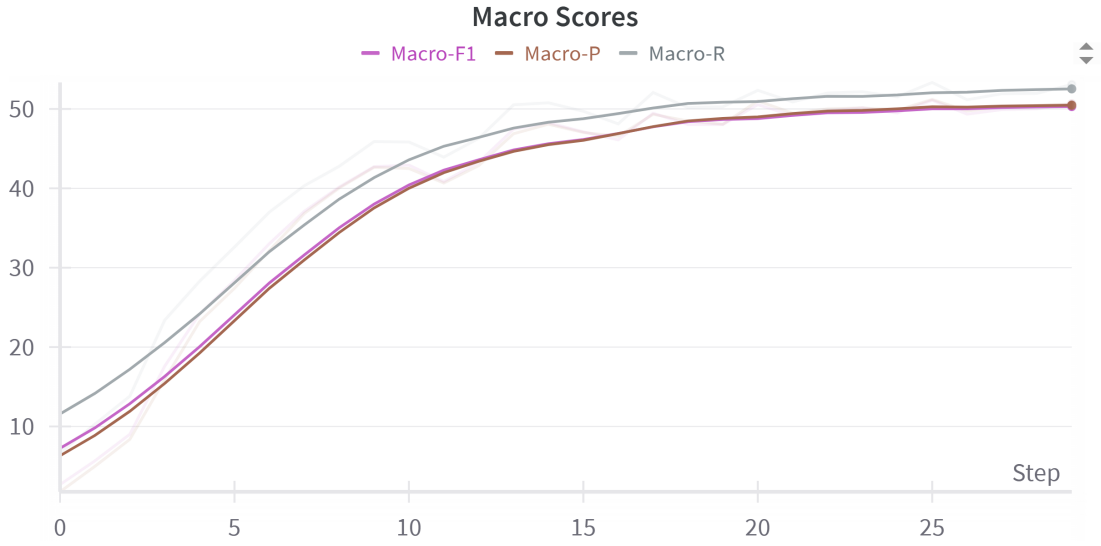
This modification resulted in a dramatic drop in performance. The Accuracy@1 fell to 26.32%, and Accuracy@5 dropped to 52.82% as seen in Fig. 11-a. The macro-averaged metrics were especially impacted, with Macro Precision at 8.65%, Macro Recall at 12.97%, and a Macro F1 Score of just 9.34%, as demonstrated in Fig. 11-b. These results suggest that without tokens, the model struggled to extract meaningful patterns from the trajectory data, leading to significantly poorer generalization and class-level discrimination.

6.4.3 Analysis

The stark contrast between the two configurations clearly establishes the importance of tokenization in the proposed model architecture. By encoding trajectory attributes into discrete, learnable units, tokens serve as a powerful medium for injecting semantic structure into the input, allowing the Transformer to better capture user-specific patterns



(a) Accuracy Graph

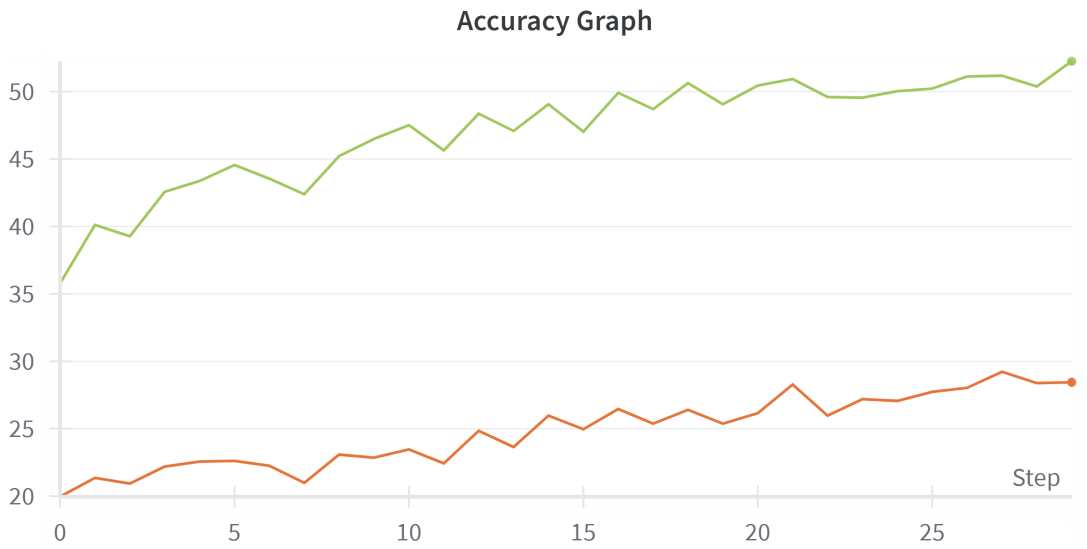


(b) Macro Scores

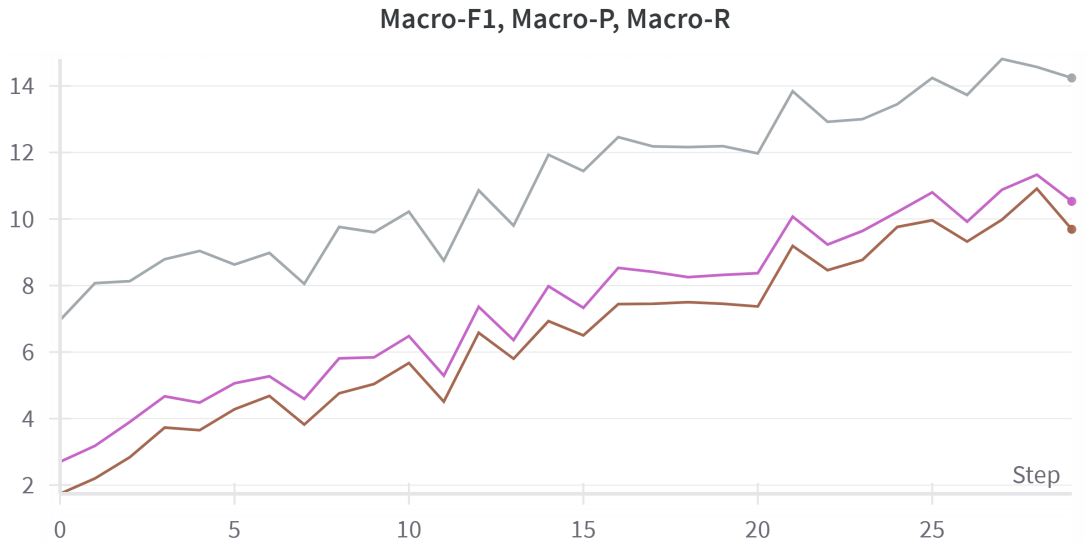
Figure 10: Performance Metrics Over Training Steps With Tokens : (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score.

and mobility characteristics. Without tokenization, the model's input space becomes less structured, making it harder for the network to differentiate between subtly varying trajectory behaviors.

Moreover, the severe degradation in macro-averaged metrics without tokens highlights that the model becomes particularly vulnerable to class imbalance and data sparsity. In contrast, the tokenized model maintains a relatively high and stable performance across



(a) Accuracy Graph



(b) Macro Scores

Figure 11: Performance Metrics Over Training Steps Without Tokens : (a) Accuracy@1 and Accuracy@5, (b) Macro Precision, Macro Recall, and Macro F1 Score.

all classes, further validating the use of token-based representations as a core component of trajectory-user linking frameworks.

6.5 Ablation Summary and Visualization

Table 6.2 presents a unified comparison of the model's performance under four configurations: the full model, a variant without the RoPE encoder, a version excluding POI information, and one without tokenization. The results clearly indicate that the removal of any individual component leads to a degradation in performance across all evaluation metrics. Notably, the exclusion of POI features and tokenization had the most drastic effect, causing a steep decline in both Accuracy@1 and macro-averaged scores. In contrast, the Transformer-only variant exhibited only a moderate performance drop, suggesting that RoPE plays a supportive—though not sole—role in the model's predictive strength.

To better illustrate these differences, Figure 12 visualizes the comparative results using a bar chart. The figure offers a more intuitive understanding of how each component contributes to the model's overall accuracy and class-level balance. Together, the table and the chart reinforce the conclusion that the synergy between structural modeling (Transformer and RoPE) and semantic enrichment (POIs and tokenization) is crucial for effective trajectory-user linking.

Table 6.2: Unified Performance Comparison: Full Model vs. Transformer Only (No RoPE), Without POI, and Without Tokenization

Metric	**Full Model**	**Transformer Only**	**Without POI**	**Without Tokens**
ACC@1 (%)	68.33	65.26	30.23	38.58
ACC@5 (%)	88.58	85.82	58.41	66.65
Macro-P (%)	51.55	48.78	12.6	17.6
Macro-R (%)	53.41	50.67	16.23	21.76
Macro-F1 (%)	51.34	48.33	13.08	18.31

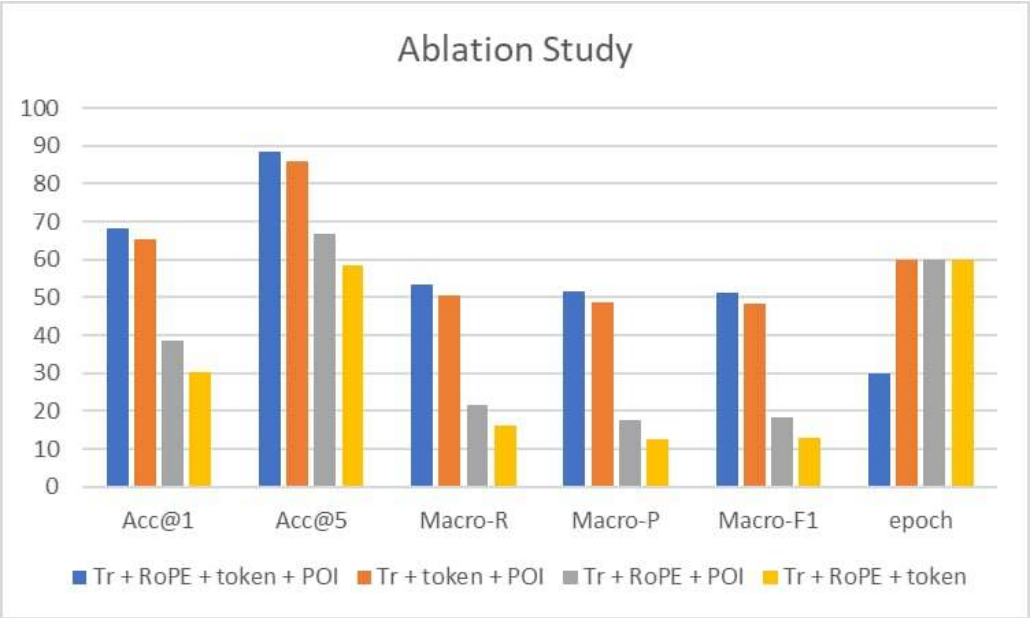


Figure 12: Bar chart visualization of ablation results for different architectural and feature configurations.

7 Conclusion and Future Works

7.1 Conclusion

In this thesis, we investigated the problem of trajectory-user linking (TUL) using spatio-temporal mobility data from the GeoLife dataset. A Transformer-based model enhanced with Rotary Positional Encoding (RoPE) and Points of Interest (POI) features was proposed to capture both the sequential and semantic properties of user trajectories. Through extensive experimental evaluation, including ablation studies and feature importance analysis, it was demonstrated that both positional encoding and semantic enrichment via POIs significantly contribute to the model's ability to accurately associate anonymous trajectories with users. The full model consistently outperformed simplified baselines, achieving strong performance across multiple evaluation metrics. The findings of this study reaffirm the importance of both temporal modeling and contextual information in tackling the challenges of trajectory-user linking tasks.

7.2 Limitations

While the results obtained are promising, the study also presents certain limitations. First, the experiments were conducted on a subset of the GeoLife dataset with trajectory lengths capped at 1000 points, which may not fully represent all real-world mobility patterns, especially those involving long or highly irregular trajectories. Second, the study focused primarily on static spatial features like POIs, without incorporating dynamic contextual factors such as weather, traffic conditions, or temporal events that could influence user movement patterns. Third, the current model's performance heavily depends on the availability and quality of POI annotations, making it less effective in regions with sparse or noisy location metadata. Finally, the experiments were limited to a few users; a broader evaluation across a larger, more diverse user base could yield deeper insights.

7.3 Ethical Considerations

The task of trajectory-user linking inherently raises important ethical concerns, particularly regarding privacy and data security. Although the research is motivated by legitimate applications such as personalized services and urban planning, the ability to re-identify users from anonymized trajectory data poses risks of misuse. It is crucial to ensure that any practical deployment of such models strictly adheres to data protection regulations, such as General Data Protection Regulation (GDPR), and incorporates mechanisms for user consent and data anonymization. Future research should also explore privacy-preserving methods, such as differential privacy or federated learning, to mitigate potential harms while retaining the benefits of mobility analytics.

7.4 Impact on Society

Accurately linking users to their mobility trajectories can have far-reaching positive impacts on society. Applications include improved location-based services, more efficient transportation planning, and enhanced public safety measures. However, the dual-use nature of such technology demands caution. While it can enable beneficial innovations, it also carries the risk of exacerbating surveillance and infringing upon individual privacy rights if misused. The societal impact of this research thus hinges on the ethical frameworks within which it is implemented, emphasizing the need for responsible AI practices.

7.5 Future Work

Building upon the findings of this thesis, several avenues for future research can be identified. One direction is to extend the model to handle longer and more complex trajectories, possibly incorporating dynamic contextual features such as time-of-day effects, social interactions, or environmental factors. Another important area is the exploration of privacy-preserving trajectory-user linking methods that balance performance with ethical considerations. Additionally, adapting the model to perform well on less structured or sparse datasets, where POI information may be incomplete or missing, represents a challenging and meaningful extension. Finally, scaling the evaluation to a larger and more heterogeneous user base, potentially across multiple cities or geographic regions, would provide a more comprehensive understanding of the model's generalizability and robustness.

*

Bibliography

[1] Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. A study of the behavior of several methods for balancing machine learning training data.

[2] Chen, C., Liu, J., Zeng, L., Duan, F., McGrath, S., and Dan, T. Spatio-temporal transformer with rotary position embedding and bone priors for 3d human pose estimation. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024* (2024), BMVA.

[3] Chen, W., Huang, C., Yu, Y., Jiang, Y., and Dong, J. Trajectory-user linking via hierarchical spatio-temporal attention networks. *ACM Transactions on Knowledge Discovery from Data 18*, 4 (Feb. 2024), 1–22.

[4] Chen, W., Li, S., Huang, C., Yu, Y., Jiang, Y., and Dong, J. Mutual distillation learning network for trajectory-user linking. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22* (7 2022), L. D. Raedt, Ed., International Joint Conferences on Artificial Intelligence Organization, pp. 1973–1979. Main Track.

[5] Cho, E., Myers, S. A., and Leskovec, J. Friendship and mobility: user movement in location-based social networks. KDD '11, Association for Computing Machinery, p. 1082–1090.

[6] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[7] Du, Y., Feng, J., Zhao, J., and Li, Y. Trajagent: An agent framework for unified trajectory modelling, 2024.

[8] Evgeniou, T., and Pontil, M. *Support Vector Machines: Theory and Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 249–257.

[9] Gao, Q., Zhang, F., Yao, F., Li, A., Mei, L., and Zhou, F. Adversarial mobility learning for human trajectory classification. *IEEE Access 8* (2020), 20563–20576.

[10] Gao, Q., Zhou, F., Zhang, K., Trajcevski, G., Luo, X., and Zhang, F. Identifying human mobility via trajectory embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17* (2017), pp. 1689–1695.

[11] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014.

[12] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural Computation 9*, 8 (1997), 1735–1780.

[13] Huang, Z., Xu, W., and Yu, K. Bidirectional lstm-crf models for sequence tagging, 2015.

[14] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98* (Berlin, Heidelberg, 1998), C. Nédellec and C. Rouveirol, Eds., Springer Berlin Heidelberg, pp. 137–142.

[15] Khandelwal, U., He, H., Qi, P., and Jurafsky, D. Sharp nearby, fuzzy far away: How neural language models use context, 2018.

[16] Kingma, D. P., and Welling, M. Auto-encoding variational bayes, 2022.

[17] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM 60*, 6 (May 2017), 84–90.

[18] Li, Y., Sun, T., Shao, Z., Zhen, Y., Xu, Y., and Wang, F. Trajectory-user linking via multi-scale graph attention network. *Pattern Recognition 158* (2025), 110978.

[19] Lin, Y., Wei, T., Zhou, Z., Wen, H., Hu, J., Guo, S., Lin, Y., and Wan, H. Trajfm: A vehicle trajectory foundation model for region and task transferability, 2024.

[20] Liu, W., Zhang, Y., Wang, J., He, Y., Caverlee, J., Chan, P. P. K., Yeung, D. S., and Heng, P.-A. Item relationship graph neural networks for e-commerce. *IEEE Transactions on Neural Networks and Learning Systems 33*, 9 (2022), 4785–4799.

[21] Miao, C., Wang, J., Yu, H., Zhang, W., and Qi, Y. Trajectory-user linking with attentive recurrent network. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Richland, SC, 2020), AAMAS '20, International Foundation for Autonomous Agents and Multiagent Systems, p. 878–886.

[22] Opitz, J., and Burst, S. Macro f1 and macro f1, 2021.

[23] Ritchie, D. M., and Thompson, K. The unix time-sharing system. *Commun. ACM 17*, 7 (July 1974), 365–375.

[24] Sherstinsky, A. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena 404* (Mar. 2020), 132306.

[25] Toohey, K., and Duckham, M. Trajectory similarity measures. *SIGSPATIAL Special 7*, 1 (May 2015), 43–50.

[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023.

[27] Xanthopoulos, P., Pardalos, P. M., and Trafalis, T. B. *Linear Discriminant Analysis*. Springer New York, New York, NY, 2013, pp. 27–33.

[28] Yang, D., Zhang, D., and Qu, B. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Trans. Intell. Syst. Technol. 7*, 3 (Jan. 2016).

[29] Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W.-Y. Understanding mobility based on gps data. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp)* (Seoul, Korea, 2008), ACM, pp. 312–321.

[30] Zheng, Y., Xie, X., and Ma, W.-Y. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin 33*, 2 (2010), 32–40. Invited Paper.

[31] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web (WWW)* (Madrid, Spain, 2009), ACM, pp. 791–800.

[32] Zhou, F., Gao, Q., Trajcevski, G., Zhang, K., Zhong, T., and Zhang, F. Trajectory-user linking via variational autoencoder. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* (7 2018), International Joint Conferences on Artificial Intelligence Organization, pp. 3212–3218.

[33] Zhou, F., Yin, R., Trajcevski, G., Zhang, K., Wu, J., and Khokhar, A. Improving human mobility identification with trajectory augmentation. *Geoinformatica 25*, 3 (July 2021), 453–483.

[34] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications, 2021.