Transferable Unsupervised Outlier Detection Framework for Human Semantic Trajectories

Zheng Zhang*
Emory University, USA
zheng.zhang@emory.edu

Hossein Amiri*
Emory University, USA
hossein.amiri@emory.edu

Dazhou Yu
Emory University, USA
dazhou.yu@emory.edu

Yuntong Hu
Emory University, USA
yuntong.hu@emory.edu

Liang Zhao
Emory University, USA
liang.zhao@emory.edu

Andreas Züfle
Emory University, USA
azufle@emory.edu

Abstract

Semantic trajectories, which enrich spatial-temporal data with textual information such as trip purposes or location activities, are key for identifying outlier behaviors critical to healthcare, social security, and urban planning. Traditional outlier detection relies on heuristic rules, which requires domain knowledge and limits its ability to identify unseen outliers. Besides, there lacks a comprehensive approach that can jointly consider multi-modal data across spatial, temporal, and textual dimensions. Addressing the need for a domain-agnostic model, we propose the Transferable Outlier Detection for Human Semantic Trajectories (TOD4Traj) framework. TOD4Traj first introduces a modality feature unification module to align diverse data feature representations, enabling the integration of multi-modal information and enhancing transferability across different datasets. A contrastive learning module is further proposed for identifying regular mobility patterns both temporally and across populations, allowing for a joint detection of outliers based on individual consistency and group majority patterns. Our experimental results have shown TOD4Traj's superior performance over existing models, demonstrating its effectiveness and adaptability in detecting human trajectory outliers across various datasets.

CCS Concepts

• **Information systems** → **Geographic information systems**; **Location based services**.

Keywords

Outlier Detection, Semantic Trajectory, Self-Supervised Learning, Geolife, Patern of Life, Simulation

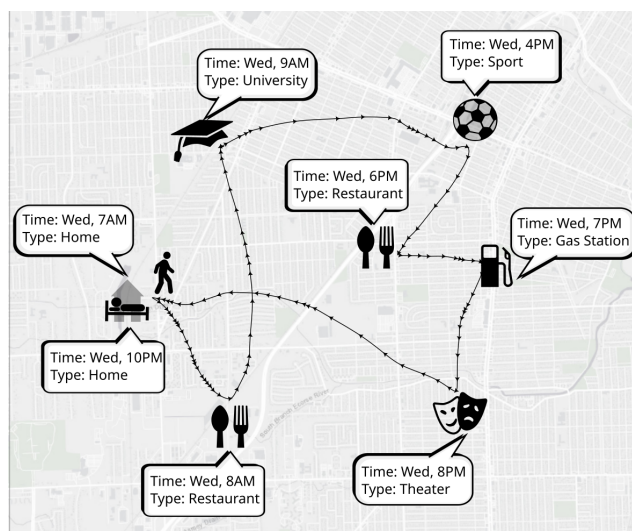*These authors contributed equally to the paper and are named in random order.

Figure 1: A semantic trajectory of a user including location trajectory and semantic information of points of interest. For each 'check-in' location in the figure, there exists an associated text description of location information.

1 Introduction

A semantic trajectory [36] is a sequence of time-ordered locations where each location is associated with a semantic label like the type of place of interest. A stylized example of a semantic trajectory is shown in Figure 1. It shows a one-day trajectory of a single user starting the day at home and visiting various places of interest (POIs) such as restaurants, a university, and recreational sites. Knowledge discovery in semantic trajectory data has been studied in the past [1] with a main focus on location prediction [48]. An important research problem that has received comparatively little attention, due to a lack of available ground truth data, is the problem of outlier detection in semantic trajectories. Yet, detecting semantic trajectory outliers may indicate a change in individual human behavior which has many important applications such as: (1) *Infectious Disease Monitoring.* A sudden change in behavior such as skipping the sports center or not going to work may indicate that a person is feeling unwell long before severe symptoms arise, infectious disease tests may detect a contagion, and even before the person is consciously aware of feeling unwell themselves. Such information may be leveraged for an early-warning system in cases where the person may have been exposed through a contact-tracing system [23, 32, 37].

(2) *Elderly Monitoring.* GPS-enabled smart-watch technology can be used to monitor the movement of elderly users [42]. In particular, if the monitored user is showing early signs of dementia, her/his trajectories could show an abrupt change from her/his movement history [44]. Detecting outliers in elder trajectories (and underlying behavior) may thus assist in early-detection and progression-monitoring of dementia.

What makes semantic trajectory outlier detection a challenging research problem is complexity of humans and their mobility [33, 34] that outliers may have many shapes and forms: Such as spatial outliers of an individual going to unusually distance POIs, temporal outliers of having individuals visit places at unusual times (such as visiting a restaurant in the middle of the night), or semantic outliers (such as an individual who does not normally drink alcohol visit a bar). An additional problem is that "one person's noise could be another person's signal" [25]. To illustrate these challenges, Figure 2 shows stylized trajectories of two example users. User 1's normal patterns of life including going to a university in addition to going to his home, nearby restaurants, and a gas station. User 2 lives in a different area and works at a courthouse. An example of a spatial outlier for User 2 may be going to a restaurant that is unusually far away. A semantic outlier for User 1 could be going to the same courthouse that User 2 works at. But since User 1 does not normally go to a courthouse, such a visit could be a deviation from the user's normal patterns of life while the same POI is normal for User 2.

Traditional methods for trajectory outlier detection [6, 7, 31, 49] predominantly rely on heuristic-based rules to identify specific types of outliers, necessitating domain-specific knowledge and limiting the detect of previously unseen outlier behaviors. Another challenge for semantic trajectory outlier detection is a lack of publicly available datasets. Commonly used (semantic) trajectory datasets such as GeoLife [55] trajectories and Location-Based Social Network Check-in Data [26] are very sparse, having very few daily trajectories for a specific region or city [21] and lack ground truth outlier labels. Therefore, an open research gap is to transfer an outlier detection model trained on a data-rich city or region (such as a simulated city) to new regions where no ground truth data is available without compromising performance. Current methodologies frequently employ manually crafted spatial-temporal features, which are usually domain-dependent and lack transfer ability across different domains.

To overcome these limitations, we introduce a <u>T</u>ransferable <u>O</u>utlier <u>D</u>etection framework for Human Semantic <u>Traj</u>ectories (TOD4Traj). This framework starts with a modality feature unification module designed to align spatial-temporal and textual data representations. This alignment facilitates the seamless integration of multi-modal data, and enhancing the model's applicability across different datasets. Additionally, we introduce a unique temporal contrastive learning module designed to represent trajectories by capturing the repetitive nature of mobility patterns. Consequently, outlier degrees are determined by considering both the consistency of an individual's behavior and the prevalent patterns among the majority. To enable other researchers to explore the field of semantic trajectory outlier detection, we make available two types of datasets for benchmarking, including a dataset obtained by systematically including outliers in the GeoLife real-world dataset, and
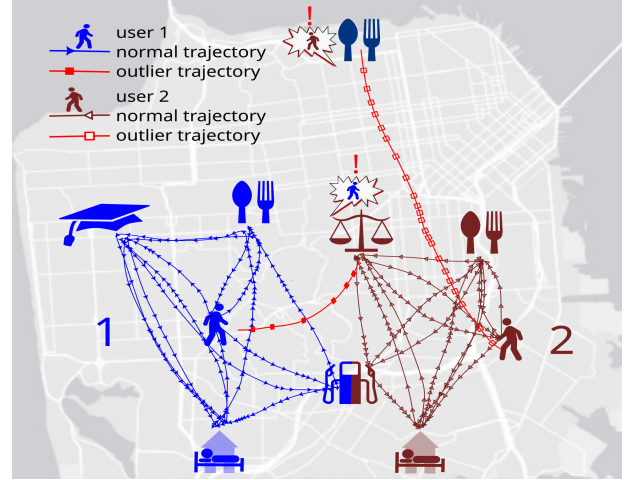

Figure 2: An example of spatial and semantic outliers. A spatial outlier for User 2 may be going to a restaurant that is unusually far away. A semantic outlier for User 1 could be going to the same courthouse that User 2 works at.

many datasets obtained through a city-level agent-based simulation of patterns of life [57]. Our experimental findings demonstrate that TOD4Traj substantially surpasses existing models in performance, thereby proving its effectiveness and adaptability in detecting outliers in varied human trajectory datasets.

In general, the contribution of this paper can be summarized into three main points. (1) We proposed a feature-level contrastive learning technique to integrate multi-modal information across spatial, temporal, and semantic dimensions; (2) A trajectory-level contrastive learning module to model the repetitiveness of human mobility patterns; (3) An outlier quantification module to simultaneously measure cross-time and cross-population abnormal behaviors. The remainder of this work is organized as follows: We begin by discussing existing human semantic trajectory outlier detection algorithms in the Section 2. This is followed by a formal problem definition and an introduction to the notations in the Section 3. Subsequently, in the Section 4, we delve into the motivation behind our approach and discuss the specific techniques employed. A thorough description of the datasets utilized in our experiments is provided next in the Section 5.1. We conclude with comprehensive experimental results, assessing aspects such as effectiveness, robustness, sensitivity, and efficiency in Section 5.

2 Related Works

Outlier detection in trajectory data. A crucial aspect of spatio-temporal data analysis, outlier detection is essential for effectively analyzing trajectory information [16, 29]. This technique has seen widespread used in a variety of fields, encompassing applications in wireless sensor networks [39, 51], climate monitoring, and transportation management [31, 46]. Surveys of traditional trajectory outlier detection algorithms can be found in [7, 31]. Important examples of such algorithms include [43] where the authors us a transfer learning approach to find outliers in areas where only a small set of trajectories are observed. In [13], the authors propose an entropy-based method designed specifically for outlier detection in scenarios where the training data contains only a few positive

instances. In [41], a real-time urban traffic outlier detection system that leverages both individual and group outlier detection was proposed. However, these approaches all aim at finding outliers in traditional trajectories defined by sequences of geo-locations without using any semantic information of the visited locations.

Contrastive learning has emerged as a promising technique in the field of unsupervised representation learning [17]. The core idea behind contrastive learning is to exploit the relationships between samples to learn meaningful representations. By contrasting positive pairs (similar samples) with negative pairs (dissimilar samples), it aims to map similar samples closer in the latent space while pushing dissimilar samples further apart. This approach obviate the need for explicit annotations or labels, making it particularly suitable for scenarios with limited labeled data. Numerous contrastive learning methods have been proposed, such as InfoNCE [35], SimCLR [10], and MoCo [19]. These methods have demonstrated impressive results in various domains, including computer vision and natural language processing, showcasing that contrastive learning as a powerful tool for unsupervised representation learning. However, the exploration of contrastive learning in the domain of semantic trajectories remains largely unexplored due to the inherent complexity and unstructured nature of trajectory data.

Semantic Trajectory Representation methods can be grouped into 1) location-level semantic information [9, 11, 52, 53] which associate each visited point of interest (or staypoint) with semantic information and 2) trajectory-level semantic information [28, 40] which associate an entire trajectory with a semantic label. Our approach uses the more general cases of location-level semantic information. Existing work on semantic trajectories has tackled important tasks such as semantic trajectory prediction [47, 48] and clustering [27]. However, to the best of our knowledge, no work has tackled the problem of finding outliers in semantic trajectories. One possible reason for the lack of existing research in this field is the lack of semantic trajectory data that includes outlier information. In this work, we fill this gap by 1) creating simulated semantic trajectory datasets where outlier information is directly included in the semantic trajectory generation, 2) providing a real-data set of semantic trajectory outliers based on the existing GeoLife [55] data, and 3) proposing a first approach towards outlier detection in semantic trajectories.

3 Preliminaries

A semantic trajectory of an individual user can be represented as a sequential list of staypoints denoted by $\mathcal{T} = \{\mathbf{p_1} \rightarrow \mathbf{p_2} \rightarrow \cdots \rightarrow \mathbf{p_n}\}$, where each staypoint $\mathbf{p_i} = (s_i; t_i; c_i)$ includes a spatial coordinate $s_i = (x_i, y_i)$, a timestamp t_i, and a semantic location class c_i. Here, n is the total count of staypoints in a trajectory. The spatial coordinates s_i specify the longitude x_i and latitude y_i positions, while the semantic class c_i identifies the type of location, such as restaurant or apartment, through descriptive text. A sub-trajectory of $\mathcal{T}$, denoted as $T^{(i,j)} \subseteq \mathcal{T}$, can be formally defined as a contiguous segment of staypoints from $\mathcal{T}$. This subset is represented as $T^{(i,j)} = \{\mathbf{p_i} \rightarrow \mathbf{p_{i+1}} \rightarrow \cdots \rightarrow \mathbf{p_j}\}$, where $1 \leq i \leq j \leq n$ and i, j are indices within the original sequence $\mathcal{T}$. This definition captures a portion of the user's trajectory, maintaining the chronological and spatial integrity of the original sequence. To encompass the collection of trajectories from multiple users, let $\mathcal{U}$ be the set of

all users. We denote the entire set of all users as a database $\mathcal{DB} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{|\mathcal{U}|}\}$, where $|\mathcal{U}|$ denotes the total number of users. Thus, each $\mathcal{T}_u \in \mathcal{DB}$ represents a sequence of semantic trajectories of a distinct user u.

Given the above definitions, here we formally formulate the semantic trajectory outlier detection problems:

Problem 1. Cross-Time Semantic Trajectory Outlier Detection. *Given a user u from the user set $\mathcal{U}$ and their set of trajectories $\mathcal{T}_u$ in database $\mathcal{DB}$, the task is to identify outlier trajectories $T_{outlier} \subseteq \mathcal{T}_u$ that exhibit significant deviation from the user's typical trajectory patterns over different time periods. These deviations are quantified using a score function f_t, which measures the degree of outlierness relative to the user's historical trajectory data.*

Problem 2. Cross-Population Semantic Trajectory Outlier Detection. *For each user $u \in \mathcal{U}$, let $\mathcal{T}_u$ represent their set of trajectories. The task involves identifying outlier trajectories $T_{outlier} \subseteq \mathcal{T}_u$ that diverge significantly from the majority pattern set $\mathcal{M}$, derived from aggregating trajectories across all users in $\mathcal{U}$. Outlier detection is based on a score function f_p, which evaluates the extent of deviation from common patterns observed across the population.*

This goal presents several unique challenges: (1) **Difficulty in seamlessly integrating multi-modal information across spatial, temporal, and semantic dimensions.** Each modality carries unique and critical information about user behaviors and patterns. Considering the interactions among data from various modalities is essential for a comprehensive identification of complex outliers. (2) **Difficulty in tracking temporal shift in user behaviors.** Human behavior is dynamic and can change due to numerous factors such as personal preferences, environmental changes, and social influences. Existing methods typically use rule-based methods, which is insufficient to handle unseen pattern shifts. Capturing these evolving patterns over time, especially in a way that accurately reflects significant shifts, demands advanced modeling techniques. (3) **Difficulty in analyzing varied user behaviors across populations.** An outlier may also occur when the trajectory pattern of an individual diverges significantly from the majority pattern observed across the broader population. The difficulty in detecting such outliers stems from the variability in behavior patterns and scalability issues with algorithms.

4 Methodology

In this section, we propose Transferable Outlier Detection for Human Semantic Trajectories (TOD4Traj) framework. Notably, our method can identify outlier behaviors without the need for labeled data. Our framework is composed of three modules: (1) To integrate the modality features across spatial, temporal and semantic dimensions, we developed a Spatial Temporal-to-Semantic contrastive learning strategy that aligns representations from disparate sources into a unified feature space, enhancing the detection of joint anomalies. Furthermore, by aligning spatial-temporal information with semantic data, we facilitate the transferability of spatial-temporal features across different datasets; (2) To effectively monitor changes in user behaviors over time, we employ a temporal contrastive learning approach that identifies the repetitiveness of human mobility patterns. This technique, produces trajectory-level embeddings, seamlessly merging spatial-temporal and semantic data from trajectory sequences. (3) To identify abnormal trajectory patterns across

users' behaviors, we have implemented a comparative analysis framework that leverages population-wide mobility trends. This framework compares individual trajectories with collective population behaviors, allowing for the detection of outliers that diverge from majority patterns.

4.1 Modality Alignment through Spatial Temporal-to-Semantic Contrastive Learning

In order to fully exploit the compatibility between semantic information and spatio-temporal information carried by the semantic information, it is crucial to enable cross-modal alignment. In this section, we developed a Spatial-Temporal to Semantic Contrastive Learning module, which aimed at integrating various data modalities. The core concept involves identifying the co-occurrence patterns within different modalities as observed in semantic spatial-temporal trajectories. Through this approach, we achieve the alignment of data across modalities into a unified, semantically-enriched, high-dimensional embedding feature space.

To effectively align spatio-temporal information with semantic information into a unified, semantically-enriched, high-dimensional embedding feature space, we utlize the co-occurrence patterns within different modalities as observed in trajectories. We adopt natural language of semantic information as supervision labels, leveraging its distinct advantages over other data sources. The primary aim of this technique is to learn a mapping that converts spatio-temporal information into natural language semantics embeddings, thereby harnessing their inherent coexistence pattern. For example, this technique aims to closely associate time-specific phrases like "Friday 6PM" with contextually relevant semantic labels, such as "entertainment place". Similarly, it seeks to connect physical locations visited with their semantic significance, enhancing the model's ability to interpret and utilize the feature embeddings meaningfully. Specifically, the Spatial-Temporal to Semantic Contrastive Learning module is designed to align the spatial-temporal representation with the semantic representation of the same trajectory point, aiming to maximize their mutual information. This process involves enhancing the similarity between spatial-temporal and semantic representations of a positive pair relative to that of negative pairs.

Formally, given a user u from the user set $\mathcal{U}$ and its set of trajectories $\mathcal{T}_u$, we explicitly link each spatial-temporal information s_i, t_i with its corresponding semantic class c_i within the same staypoint $\mathbf{p_i}$. Thus, we define the positive set $\mathcal{P}$ as:

$$\mathcal{P}(u) = \{(s_i, t_i; c_i) \mid s_i, t_i, c_i \in \mathbf{p_i}, \forall \mathbf{p_i} \in \mathcal{T}_u\}, \quad (1)$$

and the negative set contains the pairing of a spatial-temporal information s_i, t_i with the semantic class c_j from a different staypoint, which can be defined as:

$$\mathcal{N}(u) = \{(s_i, t_i; c_j) \mid s_i, t_i \in \mathbf{p_i}, c_j \in \mathbf{p_j}, i \neq j, \mathbf{p_i}, \mathbf{p_j} \in \mathcal{T}_u\}. \quad (2)$$

To encourage the similarity between the positive pairs and dissimilarity between negative pairs, we introduce the following contrastive learning objective function:

$$
\begin{aligned}
\mathcal{L}_{\text{Align}} = -\sum_{(s_i,t_i;c_i)\in\mathcal{P}(u)} & \log \frac{e^{\text{sim}(\mathbf{d_{c_i}},\mathbf{d_{t_i}})/\tau}}{\sum_{(s_j,t_j;c_j)\in\mathcal{P}(u)\cup\mathcal{N}(u)} e^{\text{sim}(\mathbf{d_{c_i}},\mathbf{d_{t_j}})/\tau}} \\
& + \log \frac{e^{\text{sim}(\mathbf{d_{c_i}},\mathbf{d_{s_i}})/\tau}}{\sum_{(s_j,t_j;c_j)\in\mathcal{P}(u)\cup\mathcal{N}(u)} e^{\text{sim}(\mathbf{d_{c_i}},\mathbf{d_{s_j}})/\tau}},
\end{aligned}
$$
$$(3)$$

where $\mathbf{d_{c_i}}$ is the embedding from the text encoder for the semantic class c_i, and $\mathbf{d_{s_i}}$ and $\mathbf{d_{t_i}}$ are the embeddings produced by the spatial-temporal encoder for the spatial-temporal information s_i and t_i, respectively. The function $\text{sim}(\cdot, \cdot)$ computes the similarity between pair embeddings from different modalities, and τ is a temperature scaling parameter that controls the separation of distributions.

In more details, a frozen pre-trained language model (e.g. BERT) is utilized to project textual information c_{ij} into a vectorized text representation: $\mathbf{d}_{c_i} = \text{PLM}(c_i)$, where $\mathbf{d}_{c_{ij}}$ is the hidden representation of the [CLS] token computed from the last layer of the PLM (pretrained language model) encoder.

To align the spatial temporal information to the semantic embeddings from PLM encoder, we develop two learnable mapping modules $\mathcal{M}_s$ and $\mathcal{M}_t$ to transform the spatial and temporal information to the semantic embedding space. Specifically, the transformed temporal embeddings can be represented as $\mathbf{d}_{t_i} = \mathcal{M}_t(t_i)$, and the transformed spatial embeddings can be represented as $\mathbf{d}_{s_i} = \mathcal{M}_s(s_i)$.

Why adopting natural language embeddings as the supervision labels? An advantage of adopting natural language supervision is because it is inherently scalable and easily interpretable by humans, unlike spatiotemporal labels. This approach also benefits from the rapid advancement in language models, enabling the generalization to unseen labels. For example, through our approach, once the model learns to associate a specific time stamp, such as 6 pm, or a defined area with certain locations like McDonald's, it gains the ability to generalize this knowledge. Consequently, the model can recognize and associate similar time or region patterns with other fast-food stores like Burger King without requiring further specific training. This method not only simplifies the learning process but also enhances the model's ability to apply learned concepts to new, yet related scenarios, thereby increasing its effectiveness in understanding and interpreting complex spatiotemporal and semantic relationships.

4.2 Modeling Semantic Trajectories through Regular Pattern Contrastive Learning

After aligning the embedding vectors to encapsulate both spatial-temporal and semantic data, we focus on developing a framework that aggregates these point-level embeddings into coherent trajectory-level representations. A key observation in human mobility patterns is the temporal consistency in an individual's activities [15]. This consistency is evident in the recurring nature of activities and mobility patterns on specific days of the week, mirroring similar behaviors on equivalent days in history. For instance, the behavior of a user on the current workday is often similar to the activities performed on previous workdays. This self-consistency extends to other days of the week as well.
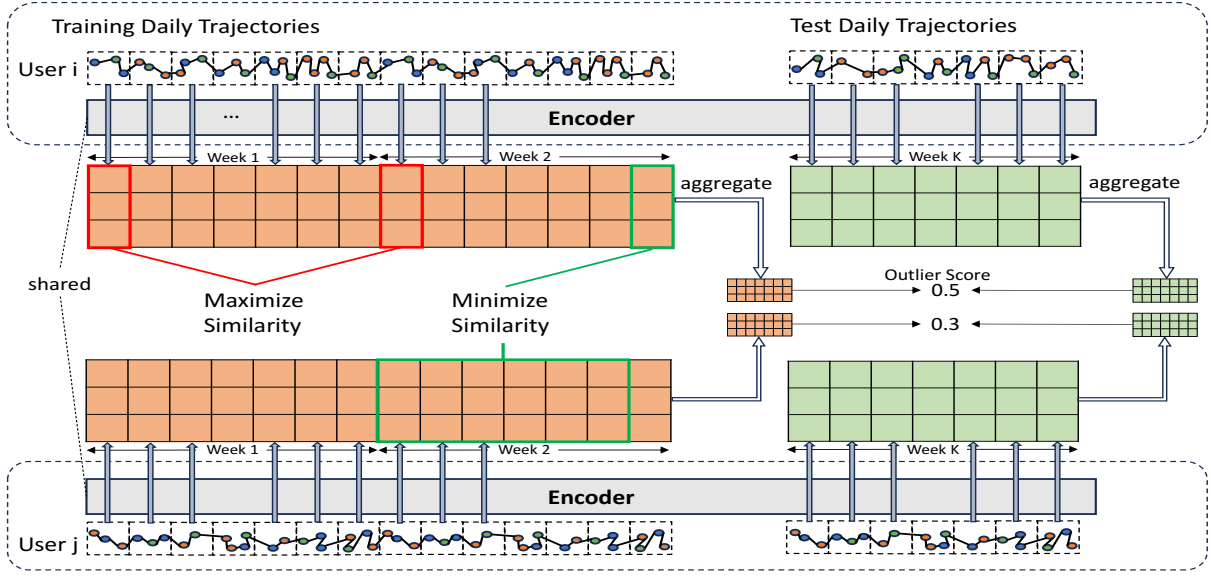
Figure 3: The illustration of our proposed model framework. (Top & Bottom) Extraction of spatio-temporal semantic embeddings from input trajectories; (Middle left) Contrastive learning based on human semantic trajectory periodicity; (Middle right) Quantification of trajectory anomaly scores based on dissimilarity between train and test embeddings.

Recognizing this consistent nature, we are inspired to incorporate two self-supervised learning tasks into our framework: (a) classifying if two daily trajectories were generated by the same user, and (b) classifying if two trajectories of the same user were generated on the same days of the week. The intricacies of this approach are depicted in Figure 3, which illustrates how the consistency in trajectories is leveraged to enhance the learning process.

Formally, large trajectories of each user are segmented into individual daily trajectories. For each user u we observe a set of trajectories $\mathcal{T}_u = [T_{u,1}, ..., T_{u,D_u}]$ where each trajectory $T_{u,d} \in \mathcal{T}_u$ corresponds to the d'th daily trajectory of the user u and D_u denotes the number of daily trajectories observed for the user i. Each daily trajectory, denoted by $T_{u,d}, d \in [1, D_u]$, is constituted by the unique locations visited by the user u within the date D_u.

To generate daily-level embeddings from distinct spatial, temporal, and semantic embeddings, we employ a deep sequential encoder $f : T \rightarrow \mathbf{z}$ to map each daily trajectory $T_{u,d} \in \mathcal{T}_u \in \mathcal{DB}$ of the user u into latent high-dimensional embeddings. This stage involves the extraction of sequentially organized spatio-temporal-semantic information from each daily trajectory, transforming them into meaningful representations. We present our model as a general framework, accommodating various commonly employed deep sequential models as potential encoders. This flexibility allows for the incorporation of classical models such as Recurrent Neural Networks (RNN) or modern architectures like Transformers. In our experimental evaluation, we explore different encoder models to verify the framework generalizability.

Finally, as illustrated in the left (green shade) of Figure 3, consider a set of daily trajectories $\mathcal{T}_u$ belonging to the user u. For a specific daily trajectory $T_{u,d} \in \mathcal{T}_u$, we construct the positive set of samples with other days corresponding to the same intrinsic pattern (e.g. a working Monday). For notational simplicity, we denote the set of

days sharing the same pattern as

$$\mathcal{D}(d) = \{d + fq | q \in \mathbb{Z} \setminus \{0\}, 1 \leq d + fq \leq D_u\}, \quad (4)$$

where f is the frequency of repeating the same pattern (e.g. $f = 7$ for a weekly repetition). Therefore, the positive pairs set $\mathcal{S}$ can be denoted as

$$\mathcal{S}(T_{u,d}) = \{T_{u,d'} | d' \in \mathcal{D}(d)\} \quad (5)$$

Conversely, for constructing negative pairs, we sample from other users and weekdays that do not align with the target day, which is denoted as

$$\mathcal{I}(T_{u,d}) = \{T_{v,d^*} | v \in \mathcal{V}, v \neq u, d^* \notin \mathcal{D}(d)\} \quad (6)$$

To operationalize the contrastive learning, without loss of generality, the objective function can be written as:

$$\mathcal{L}_{\text{Consistency}} = -\sum_{u \in \mathcal{U}} \sum_{d \in [1, D_u]} \log \frac{s_{\text{pos}}(T_{u,d})}{s_{\text{pos}}(T_{u,d}) + s_{\text{neg}}(T_{u,d})},$$

$$s_{\text{pos}}(T_{u,d}) = \sum_{T_{u,d'} \in \mathcal{S}(T_{u,d})} e^{\text{sim}(\mathbf{z}(T_{u,d}), \mathbf{z}(T_{u,d'}))/\varsigma}$$

$$s_{\text{neg}}(T_{u,d}) = \sum_{T_{u,d^*} \in \mathcal{I}(T_{u,d})} e^{\text{sim}(\mathbf{z}(T_{u,d}), \mathbf{z}(T_{u,d^*}))/\varsigma} \quad (7)$$

where $\mathbf{z}(\cdot)$ denotes the daily-level trajectory embeddings, and ς is the temperature parameter. The negative samples are chosen from the embeddings corresponding to different users and different days of the week, thereby ensuring the maximization of dissimilarity among the selected negative samples.

We acknowledge that this self-supervised approach which uses the task of classifying whether two trajectories 1) belong to the same user and 2) from the same day-of-the-week may incur confusion in for special cases such as holidays, where a holiday-Thursday may be more similar to a Sunday for some users. However, the dissimilarity between users should remain high, as different users will have different home locations, different work locations (during

work days), and different favorite locations. Thus, there should still be substantial contrast between a positive sample that suffers from confusion due to holidays and a negative sample between different users (and different days-of-the-week).

4.3 Quantification of Outlier Scores

Given the trained contrastive model that can extract human mobility pattern behavior from both spatial-temporal and semantic information, we now focus on the process of quantifying the degree of being abnormal. It is important to recognize that outliers may occur both cross-time or cross-population, as defined in Section 3. Intuitively, a cross-time outlier is indicated when a user's current trajectory pattern significantly deviates from their historical patterns. Conversely, a cross-population outlier is suggested when this pattern markedly differs from those of whole populations. This dual-focus analysis allows for a comprehensive understanding of deviations in mobility behavior, while existing methods [6, 18, 30, 49] typically only focus on one kind of the outliers.

To detect both cross-time and cross-population outliers, a straightforward approach involves comparing the user's current trajectory embedding with their past trajectory embeddings and those of other users. However, this method faces significant challenges. Measuring global mismatches comprehensively would necessitate calculating the pairwise similarity for every user pair, leading to a quadratic increase in computational complexity. To circumvent this issue, we suggest leveraging the *low-rank* properties of human mobility patterns for measuring outliers. Typically, human mobility patterns exhibit low-rank characteristics in large user sets, attributed to the regularity of human behaviors. Individuals generally adhere to a limited range of routines and visit a restricted set of locations consistently, resulting in repetitive movement patterns across a broad population. This uniformity means the entire dataset of human movements can be effectively summarized by a small set of core factors or dimensions, reflecting its low-rank nature.

Unfortunately, applying traditional low-rank techniques like Singular Value Decomposition (SVD) directly to this problem introduces generalization issues with new data, making it unsuitable for online detection methods. Furthermore, SVD demands considerable computational resources, presenting a significant challenge for efficient implementation.

To effectively harness the low-rank property within the entire dataset of human movement trajectories, we introduce a soft clustering objective into our overall training objective function. By optimizing a small set of clustering centroids, we aim to capture the essence of low-rank movement patterns. Consequently, the proximity of each trajectory to its nearest clustering centroid serves as a measure of its deviation from the mainstream patterns in the dataset. This distance becomes a crucial indicator for assessing the degree of abnormality, with greater distances suggesting more significant deviations from typical movement behaviors.

Formally, given a set of $K \ll |\mathcal{U}|$ learnable centroids $\{\mathbf{b}_k | k \in [1, K]\}$, the soft clustering objective function can be written as:

$$\mathcal{L}_{Clustering} = \sum_{u \in \mathcal{U}} \sum_{d \in [1, D_u]} \sum_{k \in [1, K]} \delta_{u,d}^{(k)} \ell(\mathbf{z}(T_{u,d}), \mathbf{b}_k),$$

$$\delta_{u,d}^{(k)} = \frac{\ell(\mathbf{z}(T_{u,d}), \mathbf{b}_k)}{\sum_{k \in [1, K]} \ell(\mathbf{z}(T_{u,d}), \mathbf{b}_k)} \tag{8}$$

where ℓ represents a distance measurement function, typically selected as $\| \cdot \|^2$. Here $\delta_{u,d}^{(k)}$ signifies the coefficient weight that allocates the current embedding $\mathbf{z}(T_{u,d})$ to the k-th centroid $\mathbf{b}_k$.

Therefore, the overall training objective can be written as:

$$\mathcal{L} = \mathcal{L}_{\text{Consistency}} + \beta \mathcal{L}_{\text{Clustering}}, \tag{9}$$

where β is a hyperparameter to balance between two terms.

Finally, the quantification of outlier scores for both cross-time and cross-population anomalies is achieved by assessing the discrepancies between (1) the historical and current patterns of an individual user, and (2) the current pattern of a user and all centroids. To achieve this, we first divide the historical trajectory data of a user into sets corresponding to each day pattern, using the notation $\mathcal{D}(d)$ as defined earlier. For each day d, we compute the average embedding of the historical trajectories as

$$\mathbf{h}_{u,d} = \frac{1}{|\mathcal{D}(d)|} \sum_{d' \in \mathcal{D}(d)} \mathbf{z}(T_{u,d'}).$$

Similarly, we compute the average embeddings $\hat{\mathbf{h}}_{u,d}$ for each day pattern from the new incoming trajectory data in the same way. Then, the cross-time outlier score for the current trajectory data can be quantified by measuring the dissimilarity between the historical embedding $\mathbf{h}_{u,d}$ and the current embedding $\hat{\mathbf{h}}_{u,d}$:

$$\text{Cross-Time}(u) = 1 - \frac{1}{f} \sum_{d \in [1, f]} \text{sim}(\mathbf{h}_{u,d}, \hat{\mathbf{h}}_{u,d}), \tag{10}$$

where f is the total number of days in the considered period (e.g., $f = 7$ for a week). Similarly, the cross-population outlier score can be quantified by measuring the dissimilarity between the current embedding with the closest centroid:

$$\text{Cross-Population}(u) = \max\{1 - \text{sim}(\hat{\mathbf{h}}_{u,d}, \mathbf{b}_k) | k \in [1, K]\}. \tag{11}$$

5 Experimental Results

We implemented all the methods, including our proposed method and competitor methods, through the Pytorch Framework. We have open-sourced all the code in the supplementary material. For a fair comparison, we require all models to follow the same experimental settings and data splits. All methods, including our proposed method and those of competitors, were implemented using the PyTorch Framework. In an effort to support transparency and reproducibility in the research community, we have provided all corresponding code at https://github.com/onspatial/transferable-outlier-detection. For a fair comparison, we maintained consistent experimental conditions across all models.

5.1 Experimental Datasets

The datasets used for this research include six simulated datasets using the Agent-Based Patterns-of-Life Simulation [2, 21, 58] and one real-world dataset based on the GeoLife dataset [55]. Specifications of the datasets, including details and key attributes, can be found in Table 1. The source code of the simulation and data processing of the GeoLife dataset is accessible through the GitHub repositories: https://github.com/onspatial/pol-outlier-dataset and https://github.com/onspatial/geolife-outlier-dataset, respectively. In addition, all datasets are available for download at https://osf.io/rxnz7/ and described in [4, 50].

5.1.1 Agent-Based Simulation of Patterns of Life . The patterns of life simulation was designed to emulate human needs and behavior in an urban environment [58]. Within the simulated environment, virtual entities referred to as agents perform actions that mirror human activities. These include attending work, forming friendships, engaging in social gatherings, and more. The agents' existence is crafted to resemble human life in a real-world environment (roads, buildings) obtained from OpenStreetMap [5, 8]. Throughout their simulated lives, agents navigate to diverse locations, including restaurants, workplaces, residential apartments, and recreational venues. A salient feature of the simulation is the generation of comprehensive log files. These logs contain extensive data regarding the agents, including their location and current state information, thus allowing for in-depth analysis and research.

In our study, we generated data by running simulations over four distinct maps, namely Fairfax County, Virginia, USA (FVA); the French Quarter of New Orleans, Louisiana, USA (NOLA); Atlanta, Georgia, USA (ATL); and Beijing, China (BJNG). The simulations were conducted over a period of 450 days to replicate normal life, followed by an additional 14 days to incorporate abnormal behavior into the regular patterns. We introduced three specific types of abnormal behavior that define outliers trajectories:

- **Hunger outlier:** An agent under this category becomes hungry more quickly. Such agents have to go to restaurants or their homes much more often.
- **Social outlier:** This type of agent randomly selects recreational sites to visit when needed, rather than being guided by their attributes and social network.
- **Work outlier:** Agents in this category abstain from going to work on workdays.

We further divided these abnormalities into three intensity levels: red, orange, and yellow. Red outliers exhibit extremely abnormal behavior, orange outliers act moderately abnormal, and yellow outliers display abnormal behavior less frequently. For example, a work outlier will decide not to go to work 100%, 50%, or 20% of the time when classified as red, orange, or yellow, respectively. We divide the simulation into 450 simulation of days of normal behavior followed by 14 days of a small number of agents exhibiting outlier behavior. Details can be found in Table 1 and, an extended version of the dataset can be found in [3].

5.1.2 Real World Dataset. The real-world dataset for this study was created using the Microsoft Research Asia's GPS Trajectory dataset [55]. Since the original data did not conform to a check-in format, we employed the method outlined in [54] to extract stay points, thereby transforming the data to fit the check-in pattern used in life simulation studies. Next, we utilized OpenStreetMap to categorize locations into four groups: apartments, workplaces, pubs, and restaurants. Given that OpenStreetMap encompasses a broad array of categories and types, we manually classified them into these four distinct groups. Upon preprocessing the data, we eliminated agents with fewer than 50 records, resulting in a final count of 69 agents with a total of 14,080 training trajectories and 3,552 test trajectories. Within the context of the GeoLife dataset, we introduced a specific outlier type called the "imposter outlier". An agent acting as an imposter outlier by switching the trajectories with another agent after a specific time point. The dataset was then

Outlier Type	#Agents	Source	Period	#Outliers
hunger	1000	POL	450+14 days	90
work	1000	POL	450+14 days	30
social	1000	POL	450+14 days	30
combined	3000	POL	450+14 days	150
imposter	69	GeoLife	4 years	20

Table 1: Detailed statistical information of the datasets utilized in this paper. Here 'POL' donotes Pattern-of-Life data.

divided into two segments: 80% of the stay points for training and introduced outliers into the remaining 20% for test.

5.2 Experimental Settings

5.2.1 Competitor Methods. We compare with several unsupervised trajectory outlier detection methods, including three rule-based non-deep learning methods and two state-of-the-art deep learning methods:

OMPAD [6] is an outlier detection method that analyzes objects' movement patterns by counting the types of locations they visit. It identifies abnormal activities by measuring the deviations in moving trends compared to established normal patterns.

MoNav-TT [49] is an outlier detection algorithm tailored for urban human trajectory networks, where it detects outliers by measuring discrepancies in traffic distances. In particular, a user is identified as an outlier if the traveled distance significantly deviates from their previous behavior.

TRAOD [25] is a partition-and-detect framework for trajectory outlier detection, which partitions a trajectory into a set of line segments, and then, detects outlying line segments for trajectory outliers.

DSVDD [38] is a deep one-class classification based outlier detection method. We generalize it to handle the task of semantic trajectory outlier detection in a most intuitive way. We map the weekly trajectories of each user to a high dimensional sphere by a deep neural network encoder. Then the distance of trajectories from the sphere's surface is quantified as an outlier score.

DAE [14, 56] is a widely-used outlier detection method that leverages a deep autoencoder. Utilizing an encoder-decoder model architecture, it reconstructs input trajectories, and the resulting reconstruction error is used as an outlier indicator, signifying deviations from the normal pattern.

GM-VSAE [30] introduces a deep generative model called Gaussian Mixture Variational Sequence AutoEncoder (GM-VSAE) for anomalous trajectory detection. GM-VSAE excels in capturing complex sequential information within trajectories, representing different types of normal routes in a continuous latent space, and facilitating efficient anomaly detection.

DeepTEA [18] is a recently proposed deep learning framework designed for time-dependent trajectory outlier detection by capturing the dynamics of traffic patterns and the temporal dependencies of movements. It uses a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to learn the normal patterns of trajectories over time. This approach allows DeepTEA to effectively identify outliers by comparing new trajectory data against learned patterns, taking into account both spatial and temporal characteristics, thus providing accurate and efficient online detection of anomalous trajectories.

	ATL				NOLA			
Model	Top-10 Hits	Top-100 Hits[*]	AP score	AUC score	Top-10 Hits	Top-100 Hits	AP score	AUC score
OMPAD	0	7	0.0571	0.5257	2	10	0.0776	0.5968
MoNav-TT	1	5	0.0893	0.4863	1	3	0.0503	0.5026
TRAOD	1	10	0.0582	0.5018	0	4	0.0485	0.5011
DSVDD	5	36	0.2601	0.5835	9	28	0.2093	0.5829
DAE	3	17	0.0962	0.5465	1	7	0.0648	0.5885
GM-VSAE	4	29	0.1987	0.5564	5	20	0.1786	0.5672
DeepTEA	5	26	0.2008	0.6012	5	26	0.2186	0.6395
Ours-MLP	**10**	34	0.2782	0.6824	**10**	41	0.3376	0.6985
Ours-RNN	**10**	32	0.2780	0.6233	**10**	27	0.2325	0.5940
Ours-CNN	**10**	**42**	**0.3205**	**0.7215**	**10**	**46**	**0.3631**	**0.7185**
Ours-Transformer	**10**	34	0.2436	0.6735	**10**	34	0.2903	0.6970
	FVA				BJNG			
OMPAD	0	4	0.0598	0.5322	1	9	0.0704	0.5655
MoNav-TT	0	0	0.0501	0.5014	1	5	0.0893	0.4863
TRAOD	0	7	0.0515	0.5090	0	6	0.0553	0.5169
DSVDD	5	26	0.2166	0.5995	10	29	0.2155	0.5643
DAE	1	7	0.0569	0.5138	0	10	0.0671	0.5568
GM-VSAE	4	22	0.1534	0.5859	4	16	0.1068	0.5479
DeepTEA	5	30	0.2221	0.6182	6	24	0.2084	0.5873
Ours-MLP	**10**	32	0.2509	0.6561	**10**	34	0.2800	0.6587
Ours-RNN	**10**	27	0.2325	0.5940	**10**	31	0.2573	0.6065
Ours-CNN	**10**	**40**	**0.3151**	**0.6669**	**10**	**66**	**0.4899**	**0.7513**
Ours-Transformer	**10**	33	0.2171	0.6628	**10**	33	0.2499	0.6219
	Geolife				ATL-Large			
OMPAD	1	4	0.1665	0.1697	3	20	0.1461	0.6028
MoNav-TT	0	7	0.2849	0.3989	1	5	0.0893	0.4863
TRAOD	4	7	0.1060	0.5498	0	1	0.0030	0.4390
DSVDD	7	15	0.6246	0.7714	1	14	0.1010	0.4911
DAE	5	12	0.4627	0.6234	4	19	0.1466	0.5641
GM-VSAE	4	13	0.4892	0.6034	2	12	0.1243	0.5482
DeepTEA	6	14	0.5290	0.7540	4	22	0.1752	0.6398
Ours-MLP	**8**	**17**	**0.8512**	**0.9397**	4	28	0.2632	0.6737
Ours-RNN	7	11	0.6359	0.7467	3	12	0.1310	0.5294
Ours-CNN	6	16	0.6756	0.8542	**10**	**40**	**0.4572**	**0.7141**
Ours-Transformer	7	16	0.6283	0.8889	8	27	0.2783	0.6852

Table 2: Outlier detection performance for all datasets. The best performance for AP and AUC scores is highlighted for each dataset. [*]We report Top-25 Hits instead of Top-100 for Geolife dataset due to its size constraint. [**]For implementation of comparison methods DSVDD and DAE, we only report the best performance of deep learning based competitive methods among the choice of four deep encoders (MLP, RNN, CNN and Transformer) for each dataset due to the limitation of the space.

	ATL				NOLA			
	Hungry	Social	Work	Total	Hungry	Social	Work	Total
Red	5 (30)	1 (10)	10 (10)	16 (50)	8 (30)	0 (10)	10 (10)	18 (50)
Orange	13 (30)	0 (10)	8 (10)	21 (50)	4 (30)	0 (10)	9 (10)	13 (50)
Yellow	3 (30)	0 (10)	2 (10)	5 (50)	7 (30)	1 (10)	7 (10)	15 (50)
Total	21 (90)	1 (30)	20 (30)	42 (150)	19 (90)	1 (30)	26 (30)	46 (150)

Table 3: Detailed detection Top-100 hits for different types of outliers and intensity levels (red, orange, and yellow denotes 100%, 50%, and 20% abnormal behavior rate over time, respectively. The outlier number in parentheses.)

5.2.2 Evaluation Metrics. To evaluate outlier detection performance, we employ the Top-K hits metrics, where the agents with the K highest outlier scores are classified as outliers. The number of hits reflects the method's ability to distinguish outliers. Specifically, we use Top-10 and Top-100 Hits to reflect the method's ability to distinguish outliers, which aligns with the size of our datasets. In addition, we utilize Average Precision (AP) and the area under the receiver operating characteristic curve (AUC) scores, which are widely used evaluation metrics for outlier detection tasks.

5.2.3 Implementation Details. Our proposed method serves as a general framework allowing for the integration of various commonly used deep representation learning techniques on trajectory data as the encoder part. To ensure a rigorous and fair comparison with competitive deep learning methods, we adopt the same deep encoders for all methods, including multilayer perceptron (MLP) [12], recurrent neural networks (RNN) [20], 1-dimensional convolutional neural networks (CNN) [24], and transformer encoder [45]. Additionally, to ensure fairness in our comparison, all deep models adhere to a uniform architecture, characterized by each daily trajectory with a cutoff length of 16, $L = 4$ encoder layers, a hidden dimension of $d = 64$, 200 training epochs, and an adaptive learning rate starting from $5e^{-3}$ with a decay rate of 0.9 for every 50 training epochs. Training is executed through back-propagation using the Adam optimizer [22], with batch sizes of 128 for regular size datasets and 32 for the ATL-large dataset. The experimental process is conducted on four NVIDIA A100-80GB GPUs.

5.3 Outlier Detection Results

5.3.1 Main Detection Results. The outlier detection performance of both our proposed method and competitive methods are presented in Table 2. We summarize the following observations:

1. The results demonstrate the superior outlier detection strength of our proposed contrastive learning method by consistently achieving the best performance across all datasets. It surpasses the second-best method with an average improvement of 0.148 in AUC scores and an additional 16.2 in Top-100 Hits. Notably, our approach achieves a perfect score, with 10 out of 10 hits in the top 10 outlier scores, on five of the six datasets.

2. We observe performance variations among different encoder choices. The 1D CNN encoder delivers the best performance in five out of six datasets, which may be attributed to its simplicity and effectiveness in extracting sequential patterns. Conversely, the RNN encoder, although outperforming most competitive methods, ranks lowest among our encoders, may be explained by its well-known issue of vanishing gradients in representing long sequences.

3. Deep learning-based methods outperform traditional ones by an average of 33.47% in AUC scores and an additional 20.42 in Top-100 Hits. This indicates that non-deep learning methods may struggle to adequately represent complex semantic trajectories, limiting their outlier detection efficacy.

4. There is worth noting that different encoder models (MLP, RNN, CNN and Transformer) exhibit relatively diversified performance. Especially, the Transformer's performance is nearly on par with CNN models, exhibiting only a 3.5% average gap. This gap in performance could be attributed to the limited amount of data available, as Transformers, with their higher number of trainable parameters, generally require more data for training. Additionally, the lower

efficacy of RNNs may due to the challenges in the optimization process, commonly referred to as the issue of vanishing gradients.

5.3.2 Detailed Detection Ratio for Types of Outlier. Besides the superior performance, it is interesting to understand what kinds of outliers and to what degree they can be detected by our algorithm. As previously mentioned that three kinds of outliers (Hunger, Social, and Work) and three abnormal intensity levels (Red, Orange, and Yellow) exist in the simulated datasets. Here, we report the detection rate of each category for Top-100 Hits in Table 3. The designed model demonstrates the ability to detect most outliers of the "Work" type but can barely detect those of the "Social" type. This may suggest that the method at recognizing location pattern changes but is less sensitive to variations in travel distances. The detection of social outliers proves to be significantly more challenging compared to the other two categories, and detecting a YELLOW level outlier is also more challenging than the other two.

5.3.3 Transfer Ability Analysis. We continue to explore the transfer capability of our proposed contrastive learning method. In real-world scenarios, there are often situations where it would be advantageous to apply a model trained on an existing dataset to a new, unseen dataset without additional training. This approach serves two primary objectives: (1) to conserve computational resources, as training a model from scratch can be both time-intensive and resource-consuming; (2) to mitigate challenges that an unseen dataset does not contain sufficient data to train a model effectively. To evaluate the efficacy of transferring our trained model to unseen datasets, we directly apply the trained model on source datasets under four transfer situations: ATL→FVA, FVA→ATL, ATL→NOLA, and FVA→NOLA, without any further adjustments. It is noteworthy that these source and target datasets comprise different user sets and different cities.

From the results in Table 4, it is evident that the transfer model, when applied from the source to the target dataset, can achieve performance on par with, or in some instances even surpassing, the model directly trained on the source dataset. Notably, when employing a CNN as the encoder, we observed even better transfer performance in three out of four cases compared to the model directly trained on the target datasets. These results demonstrate our model's effective transfer across datasets, addressing computational and data scarcity challenges.

5.3.4 Ablation Study. Here, we investigate the impact of the proposed components of our method. We consider three variants of our model *No-Semantic*, *No-Spatial* and *No-Temporal*, which remove the semantic, spatial or temporal information, separately. We report the results on ATL dataset with CNN encoder in Table 5, where the results on other datasets with other encoders are similar. Our findings indicate that every component is vital for our method's success, with performance declining upon the removal of any part.

5.3.5 Parameter Sensitivity Analysis. Here, we further conduct sensitivity analysis on the important parameters involved in our experiments. (1) We first deploy an experiment on extending the test time periods. The test period with outliers is set as two weeks in our datasets. However, it is usually difficult to determine the exact time when anomalous behavior starts. Here, we extend our test period to a longer time that includes more days before the time point anomalous behaviors start. In specific, we extend the 2 weeks test

Dataset	Encoders	Top-100 Hits	AP Score	AUC Score
ATL	MLP | original	32	0.2509	0.6561
↓	MLP | transfer	29	0.2346	0.6479
FVA	CNN | original	40	0.3151	0.6669
	CNN | transfer	41	0.3111	0.6596
FVA	MLP | original	32	0.2509	0.6561
↓	MLP | transfer	25	0.2069	0.6345
ATL	CNN | original	40	0.3151	0.6669
	CNN | transfer	38	0.3051	0.7021
ATL	MLP | original	41	0.3376	0.6985
↓	MLP | transfer	33	0.2461	0.6593
NOLA	CNN | original	46	0.3631	0.7185
	CNN | transfer	48	0.3718	0.7227
FVA	MLP | original	41	0.3376	0.6985
↓	MLP | transfer	37	0.2945	0.6622
NOLA	CNN | original	46	0.3631	0.7185
	CNN | transfer	47	0.3657	0.7134

Table 4: Transfer learning ability results for ATL to FVA and NOLA, and FVA to ATL and NOLA datasets. Here 'original' denotes training the model on target dataset from scratch, while 'transfer' denotes to apply the trained model on source dataset to target dataset without further tuning.

Category	Top-10 Hits	Top-100 Hits	AP Score	AUC Score
Full	10	42	0.3205	0.7215
No-Semantic	9	32	0.2532	0.6262
No-Spatial	10	39	0.3017	0.7000
No-Temporal	10	35	0.2908	0.6897

Table 5: Ablation studies. Comparison with the full model.

Test period	AP	ROC	# epochs	AP	ROC
2 weeks	0.3205	0.7215	50	0.2715	0.6321
4 weeks	0.3084	0.7084	100	0.2698	0.6753
6 weeks	0.2768	0.6875	200	0.2783	0.6852
8 week	0.2433	0.6675	500	0.3046	0.6554

Table 6: Parameter sensitivity analysis on comparison of test period spans and number of epochs.

period to 4,6 and 8 weeks. We report the results on ATL dataset with CNN encoder in Table 6. We can observe that the model maintains strong detection ability, with only a small drop in performance, even when extending the test period fourfold to 8 weeks. The findings demonstrate significant robustness, with a mere 7.6% decline in performance even when the noise level is quadrupled relative to the signal, which still outperforms all comparison method when no noise present. (2) We tested the model's performance with respect to the number of training epochs, varying from 50 to 500 epochs, as shown in Table 6. The results demonstrate the robustness of our model is not sensitive to parameter changes, exhibiting only a 3.16% of variation in ROC scores.

5.3.6 Efficiency Analysis. In Table 7, we present the running time per epoch at a range of training time spans from 1 month to 121 months with 1,000 agents. The results reveal that the running time for most encoders (MLP, CNN, and Transformer) follows an approximately linear growth trend with respect to the increase in training

Method	1 mon	15 mon	121 mon
MLP	0.779	11.578	110.334
RNN	0.955	18.108	614.385
CNN	0.784	10.741	113.833
Transformer	1.142	13.405	131.646

Table 7: Comparison of running time per epoch over different encoders and trajectory time spans (Unit: second).

trajectory time span. On the other hand, the slow running time with the RNN model may be attributed to its recursive structure, which can affect efficiency in large-scale parallel computing.

6 Conclusions and Future Work

In conclusion, this study advances the domain of outlier detection in human semantic trajectories by introducing a novel self-supervised learning approach that leverages the inherent temporal periodicity in human mobility behaviors. Traditional methods, which typically relied on hand-crafted spatiotemporal indicators, have been shown to possess limitations in their adaptability to unseen outlier patterns. In contrast, our methodology, built on intuitive human behavior patterns, presents a promising solution for detecting meaningful outliers of semantic trajectories. The comprehensive experiments confirmed the effectiveness, robustness, and efficiency of our proposed method. For future research directions, we are inclined to investigate the underlying factors influencing disparate model performances across various outlier types. Additionally, refining our approach to accommodate corner cases, such as holidays, may enhance the robustness of outlier detection in real-world scenarios.

7 Acknowledgements

References

[1] Luis Otavio Alvares, Vania Bogorny, et al. 2007. Towards semantic trajectory knowledge discovery. *Data Mining and Knowledge Discovery* 12 (2007).
[2] Hossein Amiri, Will Kohn, et al. 2024. The Patterns of Life Human Mobility Simulation. (2024). arXiv:2410.00185
[3] Hossein Amiri, Ruochen Kong, and Andreas Zufle. 2024. Urban Anomalies: A Simulated Human Mobility Dataset with Injected Anomalies. (2024). arXiv:2410.01844
[4] Hossein Amiri, Shiyang Ruan, et al. 2023. Massive Trajectory Data Based on Patterns of Life. In *SIGSPATIAL'23*. ACM, 1–4.
[5] Kuldip Singh Atwal, Taylor Anderson, Dieter Pfoser, and Andreas Züfle. 2022. Predicting building types using OpenStreetMap. *Scientific Reports* 12, 1 (2022), 19976.
[6] Arslan Basharat, Alexei Gritai, and Mubarak Shah. 2008. Learning object motion patterns for anomaly detection and improved object detection. In *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 1–8.
[7] Asma Belhadi, Youcef Djenouri, Jerry Chun-Wei Lin, and Alberto Cano. 2020. Trajectory Outlier Detection: Algorithms, Taxonomies, Evaluation, and Open Challenges. *ACM Trans. Manage. Inf. Syst.* 11, 3, Article 16 (jun 2020), 29 pages. https://doi.org/10.1145/3399631
[8] Jonathan Bennett. 2010. *OpenStreetMap*. Packt Publishing Ltd.
[9] Lisi Chen, Shuo Shang, Christian S Jensen, Bin Yao, and Panos Kalnis. 2020. Parallel semantic trajectory similarity join. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 997–1008.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[11] Gao Cong, Hua Lu, Beng Chin Ooi, Dongxiang Zhang, and Meihui Zhang. 2012. Efficient spatial keyword search in trajectory databases. *arXiv preprint arXiv:1205.2880* (2012).

[12] George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2, 4 (1989), 303–314.

[13] Armin Daneshpazhouh and Ashkan Sami. 2014. Entropy-based outlier detection using semi-supervised approach with few positive examples. *Pattern Recognition Letters* 49 (nov 2014), 77–84. https://doi.org/10.1016/j.patrec.2014.06.012

[14] Dario Dotti, Mirela Popa, and Stylianos Asteriadis. 2020. A hierarchical autoencoder learning model for path prediction and abnormality detection. *Pattern Recognition Letters* 130 (2020), 216–224.

[15] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *nature* 453, 7196 (2008), 779–782.

[16] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. 2014. Outlier Detection for Temporal Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 26, 9 (sep 2014), 2250–2267. https://doi.org/10.1109/TKDE.2013.184

[17] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. IEEE, 1735–1742.

[18] Xiaolin Han, Reynold Cheng, Chenhao Ma, and Tobias Grubenmann. 2022. DeepTEA: effective and efficient online time-dependent trajectory outlier detection. *Proceedings of the VLDB Endowment* 15, 7 (2022), 1493–1505.

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

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

[21] Joon-Seok Kim, Hyunjee Jin, et al. 2020. Location-based social network data generation based on patterns of life. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 158–167.

[22] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[23] Will Kohn, Hossein Amiri, and Andreas Züfle. 2023. EPIPOL: An Epidemiological Patterns of Life Simulation (Demonstration Paper). In *SIGSPATIAL SpatialEpi'23 Workshop*. ACM, 13–16.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[25] Jae-Gil Lee, Jiawei Han, and Xiaolei Li. 2008. Trajectory outlier detection: A partition-and-detect framework. In *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 140–149.

[26] Jure Leskovec and Rok Sosič. 2016. SNAP: A General-Purpose Network Analysis and Graph-Mining Library. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 1 (2016), 1.

[27] Caihong Liu and Chonghui Guo. 2020. STCCD: Semantic trajectory clustering based on community detection in networks. *Expert Systems with Applications* 162 (2020), 113689.

[28] Kuien Liu, Bin Yang, Shuo Shang, Yaguang Li, and Zhiming Ding. 2013. MOIR/UOTS: trip recommendation with user oriented trajectory search. In *2013 IEEE 14th International Conference on Mobile Data Management*, Vol. 1. IEEE, 335–337.

[29] Yueyang Liu, Lance Kennedy, Hossein Amiri, and Andreas Züfle. 2024. Neural Collaborative Filtering to Detect Anomalies in Human Semantic Trajectories. arXiv:2409.18427

[30] Yiding Liu, Kaiqi Zhao, Gao Cong, and Zhifeng Bao. 2020. Online anomalous trajectory detection with deep generative sequence modeling. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 949–960.

[31] Fanrong Meng, Guan Yuan, Shaoqian Lv, Zhixiao Wang, and Shixiong Xia. 2019. An overview on trajectory outlier detection. *Artificial Intelligence Review* 52, 4 (dec 2019), 2437–2456. https://doi.org/10.1007/s10462-018-9619-1

[32] Mohamed Mokbel, Sofiane Abbar, and Rade Stanojevic. 2020. Contact tracing: Beyond the apps. *SIGSPATIAL Special* 12, 2 (2020), 15–24.

[33] Mohamed Mokbel, Mahmoud Sakr, Li Xiong, Andreas Züfle, Jussara Almeida, Taylor Anderson, Walid Aref, Gennady Andrienko, Natalia Andrienko, Yang Cao, et al. 2022. Mobility data science (dagstuhl seminar 22021). In *Dagstuhl reports*, Vol. 12. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[34] Mohamed Mokbel, Mahmoud Sakr, Li Xiong, Andreas Züfle, Jussara Almeida, Walid Aref, Gennady Andrienko, Natalia Andrienko, Yang Cao, Sanjay Chawla, et al. 2023. Towards Mobility Data Science (Vision Paper). *arXiv preprint arXiv:2307.05717* (2023).

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[36] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, et al. 2013. Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)* 45, 4 (2013), 1–32.

[37] Sirisha Rambhatla, Sepanta Zeighami, Kameron Shahabi, Cyrus Shahabi, and Yan Liu. 2022. Toward Accurate Spatiotemporal COVID-19 Risk Scores Using High-Resolution Real-World Mobility Data. *ACM Trans. Spatial Algorithms Syst.* 8, 2 (2022), 1–30. https://doi.org/10.1145/3481044

[38] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International conference on machine learning*. PMLR, 4393–4402.

[39] Nauman Shahid, Ijaz Haider Naqvi, and Saad Bin Qaisar. 2015. Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey. *Artificial Intelligence Review* 43, 2 (feb 2015), 193–228. https://doi.org/10.1007/s10462-012-9370-y

[40] Shuo Shang, Ruogu Ding, Bo Yuan, Kexin Xie, Kai Zheng, and Panos Kalnis. 2012. User oriented trajectory search for trip recommendation. In *Proceedings of the 15th international conference on extending database technology*. 156–167.

[41] Juntian Shi, Zhicheng Pan, Junhua Fang, and Pingfu Chao. 2023. RUTOD: real-time urban traffic outlier detection on streaming trajectory. *Neural Computing and Applications* 35, 5 (feb 2023), 3625–3637. https://doi.org/10.1007/s00521-021-06294-y

[42] Thanos G Stavropoulos, Asterios Papastergiou, Lampros Mpaltadoros, Spiros Nikolopoulos, and Ioannis Kompatsiaris. 2020. IoT wearable sensors and devices in elderly care: A literature review. *Sensors* 20, 10 (2020), 2826.

[43] Yueyang Su, Di Yao, and Jingping Bi. 2023. Transfer learning for region-wide trajectory outlier detection. *IEEE Access* (2023), 1–1. https://doi.org/10.1109/ACCESS.2023.3294689

[44] Magdalena I Tolea, John C Morris, and James E Galvin. 2016. Trajectory of mobility decline by type of dementia. *Alzheimer disease and associated disorders* 30, 1 (2016), 60.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[46] Jingwei Wang, Yun Yuan, Tianle Ni, Yunlong Ma, Min Liu, Gaowei Xu, and Weiming Shen. 2020. Anomalous Trajectory Detection and Classification Based on Difference and Intersection Set Distance. *IEEE Transactions on Vehicular Technology* 69, 3 (mar 2020), 2487–2500. https://doi.org/10.1109/TVT.2020.2967865

[47] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. 2017. Serm: A recurrent model for next location prediction in semantic trajectories. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2411–2414.

[48] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S Tseng. 2011. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*. 34–43.

[49] Jianting Zhang. 2012. Smarter outlier detection and deeper understanding of large-scale taxi trip records: a case study of NYC. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*. 157–162.

[50] Zheng Zhang, Hossein Amiri, et al. 2023. Large Language Models for Spatial Trajectory Patterns Mining. (2023). arXiv:2310.04942

[51] Zheng Zhang and Liang Zhao. 2022. Unsupervised deep subgraph anomaly detection. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 753–762.

[52] Bolong Zheng, Nicholas Jing Yuan, Kai Zheng, Xing Xie, Shazia Sadiq, and Xiaofang Zhou. 2015. Approximate keyword search in semantic trajectory database. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 975–986.

[53] Kai Zheng, Bolong Zheng, Jiajie Xu, Guanfeng Liu, An Liu, and Zhixu Li. 2017. Popularity-aware spatial keyword search on activity trajectories. *World Wide Web* 20 (2017), 749–773.

[54] Yu Zheng, Xing Xie, Quannan Li, and Wei-Ying Ma. 2008. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL conference on Advance in Geographical Information Systems* (proceedings of the 16th acm sigspatial conference on advance in geographical information systems ed.). https://www.microsoft.com/en-us/research/publication/mining-user-similarity-based-on-location-history/

[55] Yu Zheng, Xing Xie, Wei-Ying Ma, et al. 2010. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* 33, 2 (2010), 32–39.

[56] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 665–674.

[57] Andreas Züfle, Dieter Pfoser, Carola Wenk, Andrew Crooks, Hamdi Kavak, Taylor Anderson, Joon-Seok Kim, Nathan Holt, and Andrew Diantonio. 2024. In Silico Human Mobility Data Science: Leveraging Massive Simulated Mobility Data (Vision Paper). *ACM Transactions on Spatial Algorithms and Systems* 10, 2 (2024), 1–27.

[58] Andreas Züfle, Carola Wenk, Dieter Pfoser, Andrew Crooks, Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, and Hyunjee Jin. 2023. Urban life: a model of people and places. *Computational and Mathematical Organization Theory* 29, 1 (2023), 20–51.