Semantic Anomaly Detection in Human Trajectories: Preserving Behavioral Patterns Through Calendar Representations

Erfan Hosseini Sereshgi
Tulane University
New Orleans, Louisiana, USA
shosseinisereshgi@tulane.edu

Mauryan Uppalapati
Tulane University
New Orleans, Louisiana, USA
muppalapati@tulane.edu

Yueyang Liu
Emory University
Atlanta, Georgia, USA
yueyang.liu@emory.edu

Lance Kennedy
Emory University
Atlanta, Georgia, USA
lance.kennedy@emory.edu

Andreas Züfle
Emory University
Atlanta, Georgia, USA
azufle@emory.edu

Carola Wenk
Tulane University
New Orleans, Louisiana, USA
cwenk@tulane.edu

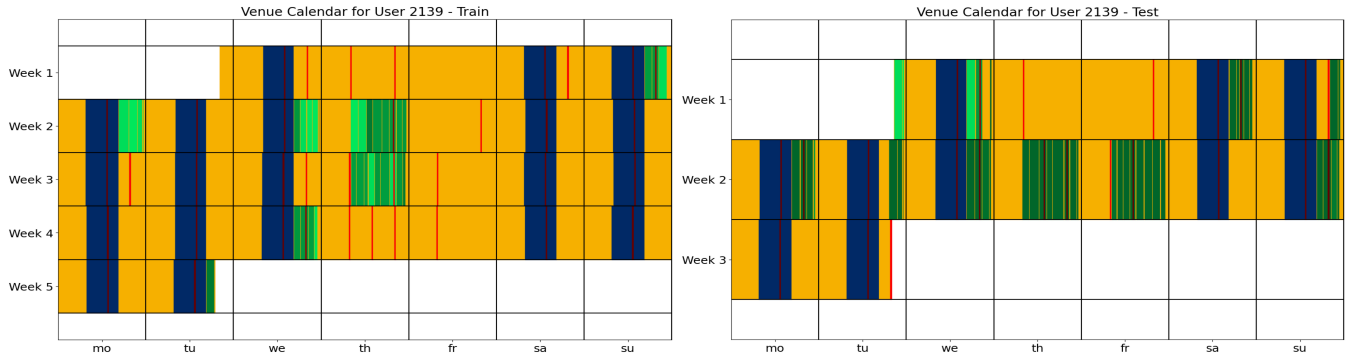


Figure 1: Visual Calendar Representations of an anomalous simulated user (agent): Rows correspond to weeks and columns correspond to five minute intervals. The colors represent the type of location visited including home (yellow), work (blue), restaurants (red), and recreation (green). Brightness denotes the distance to the user's home location. Normal behavior of the user is shown on the left and anomalous behavior is shown on the right. Can you identify the anomalous behavior of this user?

Abstract

The analysis of movement patterns has become a significant area of research largely due to the availability of new real-world and simulated datasets. A fundamental task in this domain is the analysis of movement patterns to detect anomalous behaviors, a task with critical applications ranging from preventing security threats to optimizing urban transportation planning.

This paper presents a novel methodology for detecting anomalous behaviors through the utilization of calendar representations. We explore two primary modalities for these representations: visual and textual. The visual representations encode location type and duration of activity as primary indicators, with spatial characteristics embedded through a systematic color-mapping scheme. Conversely, the textual representations prioritize spatial properties while abstracting away temporal duration. A qualitative evaluation of these calendar representations demonstrates their effectiveness in distinguishing between normal and anomalous behaviors.

Finally, we leverage the capabilities of vision-language models (VLMs) to quantitatively validate the proposed methodology. Comprehensive experiments were conducted on multiple datasets, including the real-world Geolife dataset and the simulated Patterns of Life and NUMOSIM datasets. The results, which indicate promising performance, are presented and benchmarked against state-of-the-art methods for each respective dataset.

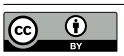
CCS Concepts

• **Information systems** → **Spatial-temporal systems**.

Keywords

Large Language Models, Trajectory data, Movement Analysis, Anomaly Detection

ACM Reference Format:
Erfan Hosseini Sereshgi, Mauryan Uppalapati, Yueyang Liu, Lance Kennedy, Andreas Züfle, and Carola Wenk. 2025. Semantic Anomaly Detection in Human Trajectories: Preserving Behavioral Patterns Through Calendar Representations. In *The 2nd ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection (GeoAnomalies '25), November 3–6, 2025, Minneapolis, MN, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3764914.3770593



GeoAnomalies '25, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2260-8/2025/11
https://doi.org/10.1145/3764914.3770593

1 Introduction

Human behavior follows rich, multi-scale patterns that define individual routines: daily commutes at 8 AM, weekly visits to religious sites on Fridays at 5 PM, monthly family gatherings, and seasonal activity shifts. However, when analyzing large-scale human trajectory data for anomaly detection, traditional methods often lose these behavioral patterns, reducing complex human routines to simple spatial coordinates or statistical features.

Finding anomalous patterns in human trajectories has a wide range of potential applications such as detecting traffic violations [15], identifying traffic accidents or jams [21], and early detection of infectious disease outbreaks through changes in human mobility [18].

Existing work on trajectory anomaly detection can be divided into two orthogonal approaches: (1) *Kinematic anomaly detection* [8, 13] asks "How do individuals travel between places?" and defines anomalies as segments that exhibit unusual speed, acceleration, or changes in direction. For example, recent work detects sharp turns and unusually slow movement [13] or detects deviations from a user's normal kinematic profile [8]. (2) *Semantic trajectory anomaly detection* asks "Where do individuals go?" and defines an anomaly as a deviation from an individual's normal patterns of life, i.e., visiting a place they should not normally visit [18] such as a child going to an arcade during school hours or an elder lost in the park at night.

In this work, we propose a new approach for semantic trajectory anomaly detection. Our approach acknowledges that "one person's noise could be another person's signal" [9]. For example, for a mailman, visiting many locations may be their normal weekly routine, while staying home on a typical workday may indicate an anomaly. Traditional trajectory analysis methods struggle to capture and preserve these nuanced individual behavioral patterns across multiple time scales.

Our approach represents the daily and weekly patterns of life of an individual as a calendar which can be transformed into a textual or visual representation. As an example, consider Figure 1, which visualizes six weeks in the life of an agent of the simulated POL dataset [17]. Four weeks of normal patterns of life for this agent are shown on the left, followed by two weeks during which the agent's decision model was alternated on the right. This calendar representation on the left immediately highlights some of this agent's patterns of life: The agent goes to work (blue color) from Saturday to Wednesday, visits many recreational sites (green) mostly on Thursdays, visits restaurants (red) at least once per day, and stays at home (yellow) all day on Fridays. On the right, we can visually observe a change in behavior: The recreation pattern has changed. The agent now visits no recreational places at all in the first anomalous week and on both Thursday and Friday in the second anomalous week. We also observe that the visited recreational sites are further away from their home (represented by the darker shades of green). The experiments described in [2] show that this type of anomaly is difficult to detect by traditional methods. This may be because the anomalous behavior of this agent could be perfectly normal for other agents. This general challenge of trajectory anomaly detection is also described in one of the pioneering trajectory anomaly detection works [9] which writes that:""one person's noise could be another person's signal". Thus, methods that seek global anomalies, which deviate from the behavior of all individuals, fail to detect these local anomalies that are anomalous only for this agent. However, Figure 1 shows how analyzing a calendar representation of an individual agent could detect anomalies.

2 Problem Definition: Semantic Trajectory Anomaly Detection

Following Zhang et al. [18], a semantic trajectory of an individual user can be represented as a sequential list of staypoints denoted by $T = \{p_1, p_2, \cdots, p_n\}$, where each staypoint $p_i = (s_i; t_i; c_i)$ includes a spatial coordinate $s_i = (x_i, y_i)$, a timestamp $t_i = (t_i^s, t_i^e)$, and a semantic location class c_i. Here, n is the total number of staypoints in a trajectory. The spatial coordinates s_i specify the longitude x_i and latitude y_i positions, the timestamp specifies the start time t_i^s and end time t_i^e, and the semantic class c_i identifies the type of location, such as a restaurant or apartment.

To encompass trajectories from multiple users, let U be the set of all users. We denote the entire database as $DB = \{T_1, T_2, \ldots, T_{|U|}\}$, where each $T_u \in DB$ represents the semantic trajectory sequence of user u.

Problem: Cross-Time Semantic Trajectory Anomaly Detection via Pattern-Preserving Representations

Given a user u from the user set U with training trajectories T_u^{train} representing normal behavioral patterns and test trajectories T_u^{test}, the task is to identify anomalous patterns in T_u^{test} that exhibit a significant deviation from the user's established patterns in T_u^{train}. These deviations are quantified using a score function $f : T_u^{test} \rightarrow [0, 100]$, where higher scores indicate greater anomaly likelihood.

This problem presents three fundamental challenges, building upon those identified by Zhang et al. [18]:

(1) **Multi-modal Pattern Integration**: Difficulty in seamlessly integrating spatial relationships, temporal structures, and semantic information while preserving behavioral patterns across multiple time scales (daily routines, weekly patterns, seasonal variations).
(2) **Representation Effectiveness**: Determining optimal encoding strategies for semantic trajectories that enable modern vision-language models to effectively detect subtle behavioral deviations and pattern shifts.
(3) **Scalable Processing**: Efficiently applying advanced AI models to large-scale trajectory datasets while maintaining detection accuracy and computational feasibility.

2.1 Key Contributions

This work makes several important contributions to semantic trajectory anomaly detection:

(1) We introduce a calendar-based representation system for semantic trajectory anomaly detection.
(2) We provide a comprehensive comparison of visual versus textual representations for trajectory anomaly detection using modern vision-language models, revealing when and why each modality excels.
(3) We evaluate our approaches and compare across diverse datasets with varying characteristics.

The combination of pattern preservation, systematic modality comparison, and comprehensive cross-dataset evaluation provides both theoretical insights and practical advances for trajectory anomaly detection.

3 Methodology

Our approach identifies anomalies through a two-phase process: we first interpret movement patterns from an agent's trajectory data and then use a vision-language model to score and flag unusual behavior.

In the first phase, described in Section 3.1, we focus on understanding an agent's typical behavior. We process trajectory data using two distinct methods to build calendar representations of agents' movement patterns. We provide more details regarding these two methods in Sections 3.1.1 and 3.1.2.

In the second phase of our methodology, described in Section 3.2, a vision-language model (VLM), specifically GPT 4.1-mini, is employed for anomaly detection. For each agent, the VLM is trained exclusively on its historical training data, which represents that agent's normal behavioral patterns. The model is then presented with the unseen test data for that same agent. The model's task is to evaluate this new data and generate an anomaly score on a continuous scale from 0 to 100, where 0 signifies normal behavior, and 100 indicates a high degree of deviation from the norm.

3.1 Movement Patterns From Trajectories

3.1.1 Generating Visual Calendar Representations. Each semantic trajectory T is transformed into a structured calendar grid $C_{visual}(T)$ where:

- **Grid structure**: Weeks form rows, days (Monday-Sunday) form columns
- **Cell content**: Each cell represents a 24-hour day containing staypoint activities
- **Color encoding**: Semantic location classes are represented by distinct colors (blue=work, yellow=home, red=restaurant, green=recreation)
- **Shading intensity**: Distance to staypoint centroid $d_i = \|s_i - \bar{s}_{c_i}\|$ encoded as color darkness (lighter=usual locations, darker=distant/unusual locations)
- **Duration encoding**: Horizontal length of color blocks represents the time spent at each staypoint

To visualize user activity patterns, we assign distinct colors to different types of locations: yellow for home, red for restaurants/food, blue for work/school, and green for social/recreation, see Figure 1.

For each individual, a central coordinate is calculated based on their training data. This "center point" represents a weighted average of all their recorded locations, with the time spent at each location serving as the weight. Consequently, locations where the individual spent more time have a greater influence on the center point's position.

This center is used to create both training and testing plots. Activities closer to this calculated center point are represented by lighter shades of their assigned color, while activities that are farther away appear in darker shades. This shading technique effectively highlights the frequency and proximity of an individual's movements in relation to their most frequented locations.

Figure 2: The color palette for the calendar plots.

For our application, a palette of four highly distinguishable RGB colors is required. A critical constraint is maintaining this perceptual separation across varying levels of brightness, as certain colors in the RGB space can exhibit a hue shift when darkened. For example, reducing the luminance of yellow $(R, G, B) = (255, 255, 0)$ can cause it to be perceived as green, complicating recognition tasks. To mitigate this, the HSV (Hue, Saturation, Value) color model is adopted. This model decouples a color's hue from its brightness, allowing the Value (V) component to be adjusted independently, thereby ensuring consistent and accurate color differentiation. The exact colors used for image generation are as follows:

- "Restaurant/Food": $(0.0, 1.0, 1.0) \rightarrow$ Red
- "Workplace/School": $(0.6, 1.0, 1.0) \rightarrow$ Blue
- "Social/Recreation": $(0.4, 1.0, 0.7) \rightarrow$ Green
- "Home/Apartment": $(0.12, 1.0, 1.0) \rightarrow$ Yellow with an orange hue

Based on these principles, we created the color palette as shown in Figure 2.

The shading function can be expressed mathematically as:

$$B(d) = B_{max} - \alpha \cdot \frac{d}{d_{max}} ,$$

where $B(d)$ is the brightness at a distance d, $B_{max} = 1.0$ is the maximum brightness, α is the brightness reduction factor, d is the current distance from the center, d_{max} is the maximum observed distance.

For most venue types, $\alpha = 0.6$, resulting in a brightness range of $[0.4, 1.0]$. For yellow/apartment venues ($h \approx 0.12$), $\alpha = 0.3$, providing a more subtle brightness range of $[0.7, 1.0]$. Finally, we convert the resulting colors to RGB to ensure compatibility with standard visualization libraries. The shading function is explained in more detail in Algorithm 1.

3.1.2 Generating Textual Calendar Representations. Each semantic trajectory T is transformed into a structured textual calendar grid $C_{textual}(T)$ where each staypoint $p_i = (s_i; t_i; c_i)$ is encoded as a three-element tuple:

$$\text{tuple}_i = (d_i, f_i, c_i) ,$$

where:

- $d_i = ||s_i - \bar{s}_c||$ maintains spatial relationships numerically, preserving the same centroid-based encoding used in the visual modality.
- f_i quantifies the historical visitation rate for location s_i based on the training dataset T_u^{train}, calculated as the normalized visit count to that specific location.
- $c_i \in \{1, 2, 3, 4\}$ represents categorical integer classification (1=Home, 2=Work, 3=Restaurant, 4=Social).

Algorithm 1: Distance-Based Color Adjustment

Input : *base_hsv*: HSV color tuple, *distance*: current distance, *max_distance*: maximum distance
Output : RGB color tuple

Function `adjust_color_by_distance(`*base_hsv, distance, max_distance*`)`**:**
  (h, s, v) ← base_hsv
  //Normalize distance to [0,1] range
  normalized_dist ← min(1.0, distance / max_distance)
  //Map normalized distance to brightness range [0.4, 1.0]
  if $|h - 0.12| < 0.01$
    //Special handling for yellow/apartment
    brightness ← 1.0 - (0.3 × normalized_dist)
  else
    brightness ← 1.0 - (0.6 × normalized_dist)
  //Ensure we stay within bounds
  brightness ← max(0.4, min(1.0, brightness))
  return *colorsys.hsv_to_rgb(h, s, brightness)*

The textual representations maintain the same calendar grid structure as the visual modality: Each day is represented as a sequence of tuples corresponding to that day's staypoint activities. While individual tuple timestamps are not included, the sequential ordering within days and across weeks maintains the temporal structure.

The encoding process is as follows:

(1) Calculate d_i using the same weighted centroid $\bar{s}_c$ from the training data
(2) For each location s_i, compute $f_i = \frac{\text{visits to } s_i \text{ in } T_u^{train}}{\text{total visits in } T_u^{train}}$
(3) Map semantic location types to integer codes using a consistent mapping across modalities
(4) Group tuples by calendar day, maintaining chronological order within each day

The beginning of an example representation is shown below:

```
Day 1: (951.56,0.0619,3) (569.68,0.4071,1)
Day 2: (569.68,0.4071,1) (1743.16,0.2566,2)
  (569.68,0.4071,1)
...
```

3.2 Vision-Language Model Integration

Our vision-language model integration employs GPT-4.1-mini in a reference-based evaluation framework that compares test period behavior against established training patterns to detect anomalous deviations in semantic trajectory representations.

3.2.1 Evaluation Framework. Our approach directly compares each agent's test period behavior against their own historical training patterns. This personalized comparison enables the detection of subtle behavioral shifts that might be normal for the population but anomalous for the individual.

For each agent u, the VLM receives both training representation R_u^{train} and test representation R_u^{test} simultaneously, where R can be either visual calendar plots $C_{visual}(T)$ or textual semantic tuples $C_{textual}(T)$. The model's task is to assess the degree of deviation between these representations and generate a quantitative anomaly score.

Our approach enables the model to recognize deviations across multiple temporal scales: Altered activity sequences within individual days, shifts in day-of-week behavioral patterns, long-term changes in activity preferences or locations, new locations or altered distance patterns from activity centers.

3.2.2 System Prompt Engineering.

Visual Modality Prompt Design. For calendar plot analysis, the system prompt provides guidance on visual interpretation:

```
You are an anomaly detection system. Your task is to
  analyze two calendar plots representing a person's
  activities and provide a single anomaly score.
Input:
You will receive two images:
Baseline Calendar (Train): Shows 28 days of activities.
  This is the normal pattern.
Evaluation Calendar (Test): Shows the subsequent 28 days
  of activities to be evaluated for anomalies.
Plot Legend:
Layout: Weeks are in rows, and days (Monday-Sunday) are
  in columns.
Activity Colors:
Blue: Work
Yellow-orange: Home
Red: Restaurant/Eating
Green: Recreation/Socializing
White: No data
Color Shades: Lighter shades mean usual, frequently
  visited locations. Darker shades mean unusual, new, or
  distant locations.
Duration: The horizontal length of a color block
  indicates the duration of the activity.
Task:
Compare the Evaluation Calendar to the Baseline Calendar.
   Based on your analysis, provide an anomaly score from
  0 to 100.
0: Indicates completely normal behavior, where the test
  patterns perfectly match the baseline.
100: Indicates highly anomalous behavior, where the test
  patterns significantly deviate from the baseline.
Anomaly Examples:
Consider the following as potential anomalies:
A significant increase in darker shades (visiting new/
  unusual places).
A significant change (increase or decrease) in the
  frequency or duration of restaurant (red) or home (
  yellow-orange) activities.
Not going to work (blue) on typical workdays.
Staying at the workplace (blue) for unusually long
  periods.
A notable increase in recreation/socializing (green)
  activities.
Output Format:
Return ONLY a single integer from 0 to 100.
Do NOT include any text, explanations, or labels.
Note: True anomalies are rare. Most cases will be normal,
   so scores above 50 should be uncommon.
Example Output:
15
```

Textual Modality Prompt Design. For semantic tuple analysis, the prompt focuses on numerical pattern recognition:

```
You are an expert in behavioral pattern analysis and
  anomaly detection. You will be given training data and
  test data for a user's location patterns.
```

```
The data consists of daily location check-ins where each
  line represents one day, and each line contains
  chronological tuples in the format (x, y, z) where:
- x: proximity to the user's typical center of activity (
  distance in meters)
- y: frequency of visiting that location (0-1 scale)
- z: venue type (1=Apartment, 2=Workplace, 3=Restaurant,
  4=Recreation)
TRAINING DATA (baseline behavior):
{train_data}
TEST DATA (behavior to evaluate):
{test_data}
Based on the training data, analyze if the test data
  shows anomalous behavior patterns. Consider:
1. Significant Changes in typical locations visited (x)
2. Significant Changes in timing patterns or order of
  locations visited comapred to the same day of the week
  in the train data
3. Significant Changes in venue type preferences (z)
4. Visiting many low proximity locations to usual
  activity centers (higher x values)
5. Visiting location with low frequency (lower y values)
Return ONLY a number between 0 and 100 where:
- 0: Completely normal behavior, no anomalies detected
- 50: Moderate anomalies, some concerning patterns
- 100: Highly anomalous behavior, significant deviation
  from normal patterns
Provide your assessment as a single number:
```

3.2.3 Scalable Processing Architecture. To address the scalability challenge identified in our problem formulation, we employ OpenAI's Batch API for large-scale processing. The batching enabled us to handle thousands of agents simultaneously and it resulted in a 50% cost savings compared to real-time API calls.

4 Datasets & Qualitative Calendar Representation Analysis

We consider the following three datasets:

- The Patterns of Life (POL) dataset [17], which is a simulated dataset using an agent-based simulation framework called the Patterns of Life Simulation [1, 3], consists of individual simulated agents that exhibit Maslowian [11] needs that trigger actions to satisfy these needs, such as going home to sleep, going to work to earn money, and meeting friends to fulfill their need for love. It injects labeled anomalies into the patterns of life by changing agents' needs.
- The NUMOSIM dataset [14], which is a simulated dataset based on an agent-based simulation designed to generate realistic human mobility trajectories across urban spaces. Each agent follows daily routines, such as commuting, shopping, and leisure activities, modeled from statistical mobility distributions. To introduce anomalies, the simulation selectively alters trajectories, e.g., by inserting unexpected detours, omitting regular activities, or modifying travel distances and durations in ways that deviate from typical patterns. These anomalies are crafted to mimic realistic irregular behaviors, such as sudden long trips, unusually short visits, or improbable travel speeds. We conduct our experiments on a subset of NUMOSIM due to its large size and the low number of anomalies.
- An augmented version of the GeoLife dataset [20], which captures the real trajectories of 192 individuals in the city of Beijing, China. Since this dataset does not include any information or labels about anomalous movements and behaviors, we include synthetic anomalies by swapping user IDs. Thus, for a specified duration, two users will swap each others locations, and we will consider these users anomalous for this duration.

Table 1 provides a brief description of the datasets. In the following, we discuss each dataset in detail and provide a qualitative analysis of using calendar image representations to identify anomalies. Then, Section 5 will provide a quantitative evaluation of the capabilities of large language (vision) models to identify anomalous semantic trajectories based on textual (visual) representations of individual trajectories.

Source	Duration	#Agents	#Anomalous Agents
POL	450 Days + 2 Weeks	3,000	150
GeoLife	4 Years	69	20
NUMOSIM	4 Weeks + 4 Weeks	200,000	381
NUMOSIM Subset	4 Weeks + 4 Weeks	1,000	50

Table 1: Specifications of the Datasets.

4.1 Patterns of Life (POL)

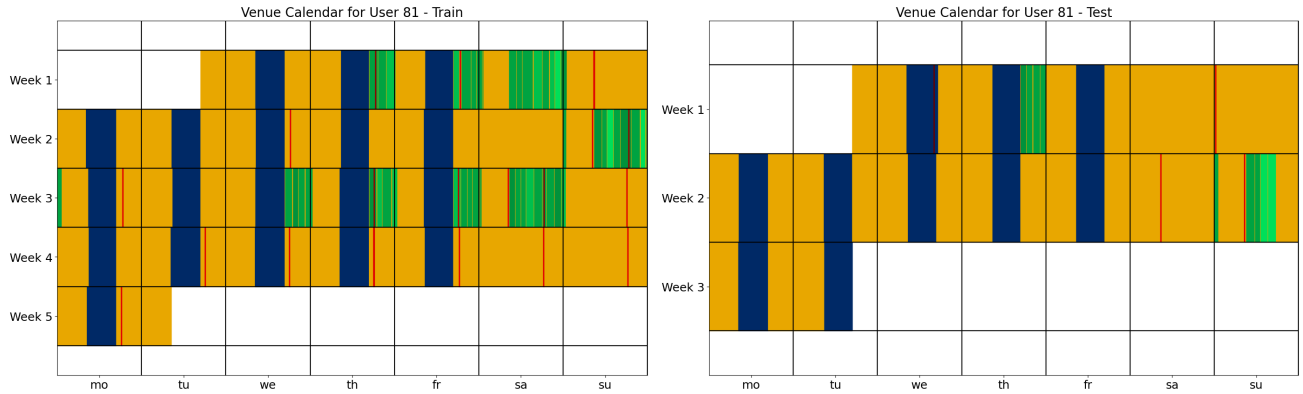
We use the simulation of Atlanta, GA, USA that was published in [17] containing 3000 individual agents during 450 + 14 days. In this dataset, the location of each agent is simulated and captured at five-minute intervals without any noise or missing data. Each point of interest that agents visit has one of four categorical semantic labels: {Home, Work, Restaurant, Recreation}. The data is provided in TSV (Tab-Separated Values) files containing check-in records with the following key fields:

- `UserId`: Unique identifier for each agent
- `CheckinTime`: Timestamp of the check-in event
- `VenueType`: Categorical classification of the venue
- `X`, `Y`: Geographic coordinates of the venue

We compute the end time for each staypoint as the `CheckinTime` of the previous staypoint. To mitigate the high computational cost and methodological inconsistencies arising from the full 450-day train dataset, the temporal window for the training data was reduced to the last four weeks. This ensures uniformity across all datasets used in the study.

An example visual calendar representation is shown in Figure 1 which shows a "Social Anomaly" that we discussed in Section 1. Figure 3 additionally shows a "Hunger" anomaly, which, for these particular agents, exhibits a *reduced* need for food. We observe in Figure 3a that this agent visits restaurants (red color) on most days, sometimes even twice a day. During the anomalous period for which the calendar representation is shown in Figure 3b we observe that the agent now rarely visits restaurants, with some of the few visits occurring at unusual times, such as around midnight between Saturday and Sunday.

Comparing normal and anomalous behavior between the calendar representations of Figure 1 and Figure 3, it seems that we can qualitatively infer an anomalous change in the agents' patterns of life. This result is promising, as traditional methods compared



(a) Normal Behavior **(b) Hunger Anomaly**

Figure 3: The visual calendar representation of an agent in the POL dataset that is labeled as a "Hunger Anomaly". This agent has a reduced food-need which causes the agent to visit fewer restaurants (red) during the anomalous periods, and visits the restaurants at untypical times.

in [18] struggle to identify anomalies. We feel that the traditional methods, which interpret a semantic trajectory as a sequence rather than a calendar, fail to identify daily and weekly patterns, such as going to work at a daily frequency (every 288 five-minute steps) or visiting a recreational site every Saturday (every 2016 five-minute steps). In contrast, the calendar representation allows us to easily discern weekly patterns as vertical bars and daily patterns as recurring patterns within a weekly row. This qualitative result raises hope that a large vision model trained on this visual calendar representation may be able to discern anomalies. A quantitative analysis to answer this question is found in Section 5.

4.2 NUMOSIM

The NUMOSIM dataset [14] contains human mobility simulation data for the Los Angeles region, representing synthetic agent movements and activities over time. The dataset is stored in Parquet format for efficient data handling and consists of five main files, including the staypoints or Point of Interest (POI) data, train data, test data, ground truth data, and demographics. The POI file contains the following information:

- `poi_id`: Unique identifier
- `name`: Name/description of the staypoints
- `latitude`: Geographic latitude coordinate
- `longitude`: Geographic longitude coordinate
- `act_types`: List of activity types

Table 2 shows the 16 activity types. Staypoints typically have multiple activity types. For example, a restaurant might have `[2, 7]` indicating that both work and eating activities can be performed at this place. The large number of activity types (16) presents a significant challenge for visualization, as creating a corresponding set of perceptually distinct and combinatorially recognizable colors is infeasible. To ensure visual clarity, a data consolidation strategy is implemented. The 16 activity types are mapped to a more constrained set of 4 primary categories, aligning the classification scheme with that of two other datasets utilized in this study. Table 2 also shows this mapping. As part of the consolidation, a data preprocessing step is performed to eliminate redundancy. Three

Code	Activity Type	New Type	New Code
0	Transportation	-	-
1	Home	Apartment	1
2	Work	Workplace	2
3	School	Workplace	2
4	ChildCare	Workplace	2
5	BuyGoods	Restaurant	3
6	Services	Restaurant	3
7	EatOut	Restaurant	3
8	Errands	Restaurant	3
9	Recreation	Recreation	4
10	Exercise	Recreation	4
11	Visit	-	-
12	HealthCare	Recreation	4
13	Religious	Recreation	4
14	SomethingElse	Recreation	4
15	DropOff	-	-

Table 2: Activity Type Definitions

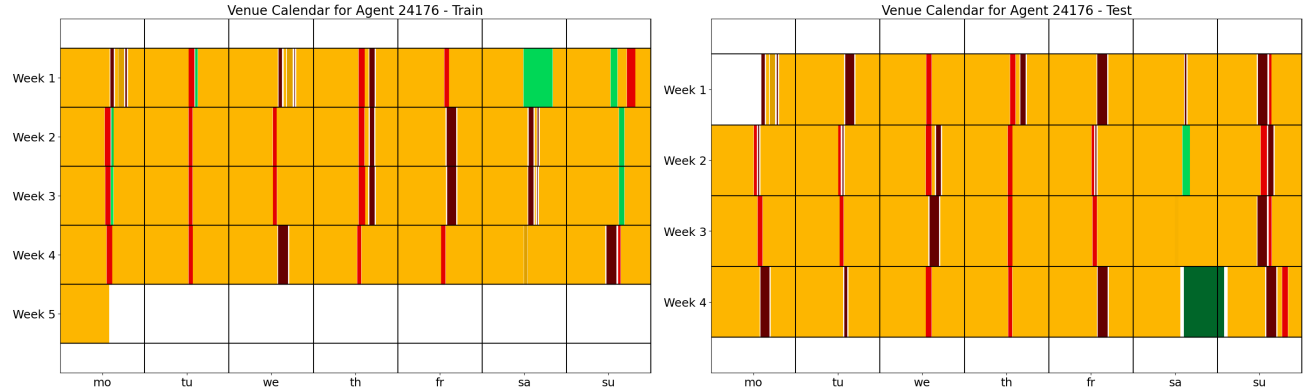
activity types, `Transportation`, `Visit` and `DropOff`, are excluded from the analysis because they consistently co-occur with other, more descriptive tags, rendering them redundant for our purposes.

To resolve cases where a single staypoint is associated with multiple activity types, a deterministic disambiguation rule is applied. The activity type with the highest numerical code is selected as the definitive classification for the staypoint. An exception is made for the `Something Else` category, which is systematically excluded from this selection criterion to prioritize more specific classifications.

The train, test, and ground truth data files contain the columns:

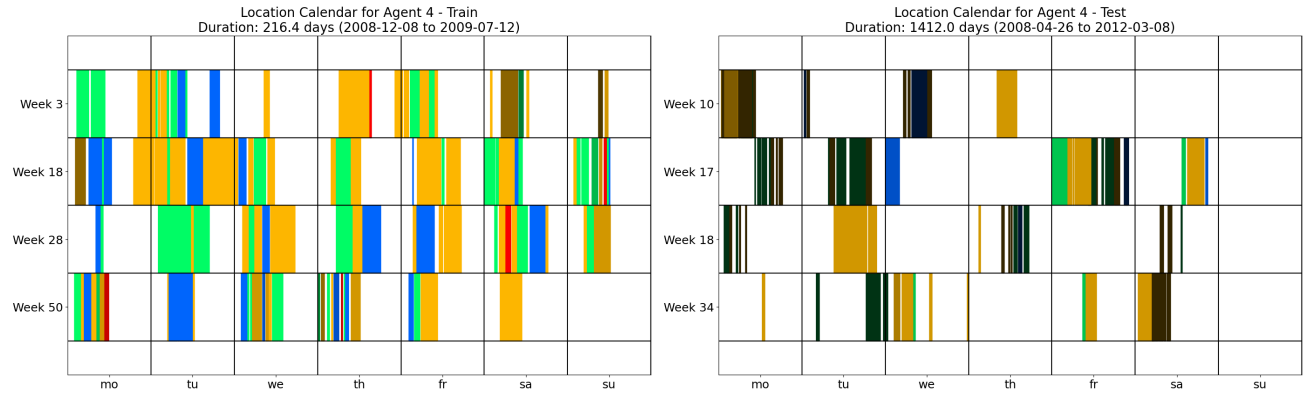
- `agent_id`: Agent identifier
- `poi_id`: staypoint identifier where the agent stayed
- `start_datetime`: Start time of the stay
- `end_datetime`: End time of the stay

Projecting this data to a specific agent, and joining the POI categories on `poi_id` allows us to create a visual calendar representation of an agent's mobility, as shown in Figure 4 for Agent



(a) Normal Behavior **(b) Anomalous Behavior**

Figure 4: The visual calendar representation of Agent 24176 in the NUMOSIM dataset.



(a) Normal Behavior **(b) Anomalous Behavior**

Figure 5: The visual calendar representation of Agent 4 in the GeoLife dataset.

24176. Investigating the normal behavior of this agent shown in Figure 4a we observed, compared to the POL data, a less stringent work routine; this agent seems to not work at all. However, we do observe a consistent pattern of visiting restaurants almost daily at nearly the same time. Figure 4b shows a month of data in which this agent exhibits anomalous behavior. We note that, unlike POL, where the agent behaves anomalously (due to a change in their causal needs) during the entire anomalous period, NUMOSIM generally has only a small number of anomalous trips inserted into an otherwise normal trajectory. Searching for anomalous patterns not seen in Figure 4a, we observe that during the test duration, the agent stops to go to recreational sites on Sundays. We also observe that the agent visits a recreational site, indicated by a green color with a darker shade, thus indicating a long distance to the agent's centroid. Furthermore, the length of this act is long, indicating that the agent stayed for an unusually extended period compared to their normal activities. Finally, the ground truth data confirms our suspicion that the agent was indeed anomalous because of that visit.

4.3 GeoLife

This dataset tracks real-world users in Beijing, China, over a period of several years. The data is provided in CSV (Comma-Separated Values) files, separated by user, containing records with the following key fields:

- `Latitude`, `Longitude`: Geographic coordinates of the user
- `Date`, `Time`: Timestamp at which this location was recorded

Since this data lacks specified check-ins, Zhang et al [18] applied a staypoint extraction algorithm [19] to create them and used a geocoding solution to map the extracted staypoints to the nearest points of interest using OpenStreetMap data. Based on this mapping, they categorized the check-ins as apartments, workplaces, restaurants, or recreational sites, similar to POL data. After removing agents with insufficient data, the final dataset consists of 69 agents over a period of four years. They then divided the dataset so that 80% of check-ins were used for training and 20% for testing, and they introduced anomalies to the dataset by swapping the check-ins of two different agents in the test portion, creating 'imposters'.

The final stage of the data contains the following columns:

- `Latitude`, `Longitude`: Geographic coordinates of the user
- `ArrivingTime`: Start time of the stay
- `LeavingTime`: End time of the stay
- `LocationType`: Categorical classification of the venue
- `AgentID`: Agent identifier

The dataset presents two primary limitations: a lack of uniform duration in the train and test periods across agents, and sparse data with significant temporal discontinuities. To normalize the input data and maintain consistency with other datasets in this study, the analysis was confined to the four weeks with the highest recorded activity within both the training and testing periods.

An example of normal and anomalous agent behaviors is presented in Figure 5. First, we observe sparsity: More often than not, we do not know the location of the agent (indicated by white color). We also see that it is quite difficult to discern any recurrent patterns of life as we were able to see in the POL dataset shown in Figure 3. For example, we do not observe any patterns of coming home at night or working on any specific schedule. The lack of home patterns could be explained by GeoLife users turning off their location tracking at night, such that the missing data may often indicate being at home. We still observe some visits to apartment-type places near the user's centroid (bright yellow), which might be home stays during which the user did not turn off their location tracking, and distant apartment-type visits (dark yellow) which might indicate visiting other people. The lack of work-type visits (blue) seems to indicate that this user may not work or may turn off their location tracking during work hours. We do see a fairly frequent commute to a nearby recreation-type place (bright green) that may indicate some pattern of normalcy. Comparing this to the calendar representation of anomalous behavior shown in Figure 5b, in which the identity of the user was swapped with another 'impostor'. We immediately observe much darker colors indicating that this user frequents places that are far from the centroid location of the user's normal behavior. This makes sense, since this is a different user who follows different patterns of life and may live in a different part of Beijing. We also note that this user is substantially more sparse, and we have an even harder time discerning any recurring patterns of life. But as a takeaway, the darker colors indicate a spatial shift in patterns of life that we can discern quantitatively and which a large vision model may be able to detect automatically.

5 Experimental Results

While the previous section provided a qualitative analysis showing that a visual calendar representation allows a human interpreter to detect anomalies qualitatively, this section investigates quantitatively whether large language and large vision models can detect anomalies. The code used to generate the image and text-based calendars is publicly available at https://github.com/Erfanh1995/CalendarAnomalyDetection, and the resulting datasets are accessible at http://doi.org/10.17605/OSF.IO/ZA84P.

5.1 State-of-the-Art Comparison

Our experimental results demonstrate highly competitive performance compared to existing trajectory anomaly detection methods across multiple datasets. Table 3 presents a comprehensive comparison with state-of-the-art approaches on the GeoLife dataset, Table 4 on the POL dataset, and Table 5 on the NUMOSIM dataset.

Performance on Geolife Data. On the GeoLife dataset (Table 3), we observe a negative result: GPT-4.1-mini is not able to find anomalous trajectories using the visual calendar representation better than state-of-the-art methods using our prompting strategy described in Section 3.2. The resulting Receiver Operating Characteristic Area Under the Curve (ROC-AUC, denoted as AUC in the tables) is 0.5683, which is not substantially better than random guessing (which would have a ROC-AUC of 0.5), and this result is substantially worse than state-of-the-art methods which achieve a ROC-AUC of up to 0.9397. This seems to indicate that, without any fine-tuning on examples of anomalous and non-anomalous train-test pairs, GPT-4.1-mini does not seem to understand the visual signal that seems obvious to a human observer. However, as a positive result, we observe that the text-based representation approach described in Section 3.1 achieves competitive results with state-of-the-art methods yielding an AUC of 0.9158 on GeoLife, performing close to the state-of-the-art TOD4Traj method (AP: 0.8512, AUC: 0.9397) with only a 6.8% gap in AP and a 2.5% gap in AUC. Notably, our approach outperforms most other state-of-the-art solutions and achieves this performance using only an existing vision-language model with carefully designed system prompts via API calls, whereas TOD4Traj requires extensive feature engineering and complex model architectures.

Performance on POL Data. On the POL dataset, we again observe a negative result for the vision-based calendar representation, yielding results that are worse than state-of-the-art. But again, we observe a positive result for our text-based method, achieving an average precision (AP) of 0.2130, which is much higher compared to the next-best baseline ([17]) AP of 0.0365, representing a 483% improvement, even though the AUC was worse. This asymmetric result can be explained by looking in more detail at the distribution of anomaly scores in Figure 6. It shows that there are a large number of true positives (red, high scores), explaining the high AP, but also a large number of false negatives (red, low scores); and we have a low number of false positives (blue, high scores). This explains our high accuracy among the Top-10 and Top-25 results also reported in Table 4, which substantially outperforms the state-of-the-art.

Performance on NUMOSIM Data. Table 5 shows preliminary results on the NUMOSIM dataset [14]. First, we note that we were only able to process a subset of the NUMOSIM data having $1,000$ of $200,000$ of the agents and including 50 of 381 anomalies due to the monetary cost of API calls to GPT-4.1.-mini's vision language model. Running the prompts described in Section 3.2 for 1000 agents costs about US$4, and thus would have cost close to US$1000 for the full dataset. In this subset, the density of true positives is much higher, thus making the problem of finding anomalies much easier. For example, random guessing on the full dataset would yield an expected average precision of $381/200,000 = 0.001905$ whereas in our set, random guessing would yield an expected average precision of $50/1000 = 0.05$. To illustrate that these results are not directly comparable, Table 5 displays our results separately. We observe a higher average precision than the results reported in [14] but we attribute

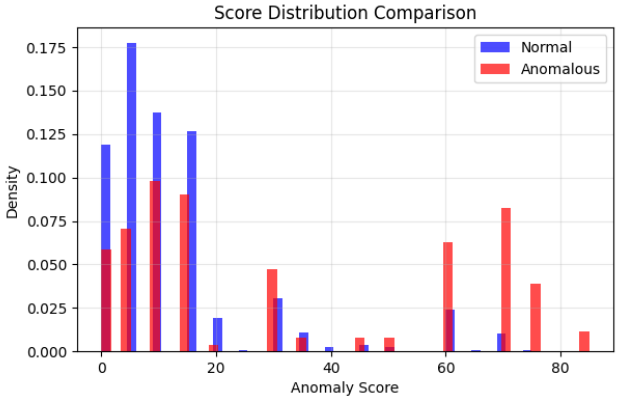


Figure 6: Score distribution of the POL (text) data.

	Geolife			
Model	Top-10	Top-25	AP	ROC-AUC
OMPAD [4]	1	4	0.1665	0.1697
MoNav-TT [16]	0	7	0.2849	0.3989
TRAOD [9]	4	7	0.1060	0.5498
DSVDD [12]	7	15	0.6246	0.7714
DAE [7, 22]	5	12	0.4627	0.6234
GPT-3.5[17]	5	8	0.4014	0.4979
Claude[17]	4	13	0.4756	0.7474
TOD4Traj[18]	**8**	17	**0.8512**	**0.9397**
Ours (image)	4	8	0.3668	0.5683
Ours (text)	**8**	**18**	0.7936	0.9158

Table 3: Comparison with the state-of-the-art on GeoLife. Results for competitor approaches are taken from [17] and [18].

	Patterns-of-Life			
Model	Top-10	Top-100	AP	ROC-AUC
OMPAD [4]	0	0	0.0079	0.4512
MoNav-TT [16]	0	0	0.0094	0.4798
TRAOD [9]	0	1	0.0030	0.4390
DSVDD [12]	1	2	0.0120	0.5398
DAE [7, 22]	0	1	0.0089	0.4649
GPT-3.5[17]	0	6	0.0365	**0.7572**
Ours (image)	2	3	0.0588	0.5561
Ours (text)	**8**	**35**	**0.2130**	0.6944

Table 4: Comparison with the state-of-the-art on POL. The results for competitor approaches are taken from [17].

this improvement to the smaller dataset and the higher fraction of true positives. Interestingly, we observe that for NUMOSIM, our vision-based approach outperforms the text-based approach, which presents a different pattern from what we observed on GeoLife and POL.

	NUMOSIM	
Model	AP	ROC-AUC
RioBusData [5]	0.001	0.501
STOD [6]	0.001	0.518
GM-VSAE [10]	0.001	0.507
Visit Rate [14]	**0.016**	**0.646**

	NUMOSIM Subset	
Ours (image)	**0.085**	**0.661**
Ours (text)	0.071	0.628

Table 5: Comparison with the state-of-the-art on NUMOSIM. Results of competitors are taken from [14].

5.2 Cross-Modal Performance and Dataset Characteristics

Our evaluation across three datasets reveals that modality effectiveness is fundamentally driven by dataset characteristics, particularly data density, temporal structure, and types of anomalies.

The density of staypoint patterns emerges as the primary determinant of visual modality effectiveness. NUMOSIM's rich behavioral data, with agents following structured daily routines across four weeks of training and four weeks of testing, enables clear visual pattern establishment in calendar representations. This density allows the image-based approach to achieve superior performance (AP: 0.085 vs 0.071 text), as the VLM can effectively recognize behavioral deviations through a visual comparison of well-established patterns.

Conversely, GeoLife's extremely sparse data presents a fundamentally different challenge. With real-world trajectory data spanning four years but containing significant temporal discontinuities, the visual calendar representations show "single digit stains across the whole train period", creating challenges for meaningful visual pattern formation due to missing data. This sparsity severely hampers the image modality (AP: 0.367), while the text-based approach excels by leveraging semantic tuple encoding to capture the limited available information more effectively (AP: 0.794).

Temporal Window Requirements for Effective Comparison. The temporal structure of train/test periods significantly affects cross-modal performance patterns. NUMOSIM's balanced four-week training and four-week testing structure provides optimal conditions for both modalities, with sufficient data for pattern establishment and adequate test duration for anomaly detection.

POL's asymmetric temporal structure, having four weeks of dense training data followed by only two weeks of testing, creates a modality-dependent performance gap. While the training period contains rich behavioral patterns (agents following Maslowian needs with five-minute interval recordings), the limited two-week test window may not provide sufficient visual evidence for reliable pattern comparison. This temporal constraint favors text-based analysis, which can effectively process semantic relationships even with shorter evaluation periods, resulting in our text approach's remarkable 483% improvement over state-of-the-art (AP: 0.213 vs 0.0365) while image performance remains modest (AP: 0.053).

Anomaly Type and Representation Alignment. The fundamental nature of anomalies determines the optimal representation choice,

revealing a clear alignment between anomaly characteristics and modality strengths. GeoLife's identity-switched agents create semantic inconsistencies perfectly suited for tuple-based text encoding. When agents swap locations, the resulting patterns exhibit systematic deviations in distance-frequency-activity combinations that text representations can precisely capture, achieving a substantial 27% AP improvement over the best traditional method.

POL's behavioral anomalies, induced by altered Maslowian needs (such as "Hunger Anomaly" reducing restaurant visits or the "Social Anomaly" changing recreational patterns), represent changes in established behavioral routines. While these patterns are theoretically detectable through visual comparison, the combination of limited test duration and the subtle nature of need-based behavioral shifts creates challenges for visual recognition.

NUMOSIM's trajectory-level anomalies—including unexpected detours, modified travel patterns, and unusual activity sequences—are well-suited for visual detection when sufficient pattern density exists. The simulation's insertion of anomalous trips into otherwise normal trajectories creates visual discontinuities that calendar representations can effectively highlight, explaining why image modality slightly outperforms text on this dataset.

6 Conclusion and Discussion

We introduced a novel approach to semantic trajectory anomaly detection that preserves behavioral patterns through calendar-based representations, addressing a fundamental limitation of traditional coordinate-based methods that lose multi-scale temporal and behavioral context. Our evaluations across three diverse datasets show that pattern preservation significantly enhances anomaly detection capabilities while revealing important insights about modality selection and dataset characteristics. Additionally, our results reveal the significant potential of vision-language models for trajectory analysis in the near future as these models evolve. Although, for large datasets, the cost factor might highlight the benefits of the textual approach. Our initial model already performs well with little tuning. We expect that by training specialized models to detect movement anomalies, our approach will outperform the best current methods while remaining straightforward to implement.

Acknowledgments

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number 140D0423C0025. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

We thank Elena Yang and Austin Nguyen for the initial prototype code for the calendar plots.

References

[1] Hossein Amiri, Will Kohn, Shiyang Ruan, Joon-Seok Kim, Hamdi Kavak, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. 2024. The Patterns of Life Human Mobility Simulation. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems* (Atlanta, GA, USA) *(SIGSPATIAL '24)*. Association for Computing Machinery, New York, NY, USA, 653–656. doi:10.1145/3678717.3691319

[2] Hossein Amiri, Ruochen Kong, and Andreas Züfle. 2024. Urban Anomalies: A Simulated Human Mobility Dataset with Injected Anomalies. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*. Association for Computing Machinery, New York, NY, USA, 1–11.

[3] Hossein Amiri, Shiyang Ruan, Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Zufle. 2023. Massive Trajectory Data Based on Patterns of Life. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems* (Hamburg, Germany) *(SIGSPATIAL '23)*. Article 49, 4 pages. doi:10.1145/3589132.3625592

[4] 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.

[5] Aline Bessa, Fernando de Mesentier Silva, Rodrigo Frassetto Nogueira, Enrico Bertini, and Juliana Freire. 2016. Riobusdata: Outlier detection in bus routes of rio de janeiro. *arXiv preprint arXiv:1601.06128* (2016).

[6] Michael Cruz and Luciano Barbosa. 2020. Learning GPS point representations to detect anomalous bus trajectories. *IEEE Access* 8 (2020), 229006–229017.

[7] 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.

[8] Lance Kennedy and Andreas Züfle. 2024. Kinematic Detection of Anomalies in Human Trajectory Data. arXiv:2409.19136 [cs.LG] https://arxiv.org/abs/2409.19136

[9] 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, IEEE, Washington, DC, 140–149.

[10] 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.

[11] Abraham Harold Maslow. 1943. A theory of human motivation. *Psychological review* 50, 4 (1943), 370.

[12] 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, PMLR, 4393–4402.

[13] Wei Shao, Ziquan Fang, Lu Chen, and Yunjun Gao. 2025. Towards Trajectory Anomaly Detection: a Fine-Grained and Noise-Resilient Framework. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. Association for Computing Machinery, New York, NY, USA, 2490–2501.

[14] Chris Stanford, Suman Adari, Xishun Liao, Yueshuai He, Qinhua Jiang, Chenchen Kuai, Jiaqi Ma, Emmanuel Tung, Yinlong Qian, Lingyi Zhao, Zihao Zhou, Zeeshan Rasheed, and Khurram Shafique. 2024. NUMOSIM: A Synthetic Mobility Dataset with Anomaly Detection Benchmarks. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection* (Atlanta, GA, USA) *(GeoAnomalies '24)*. 68–78. doi:10.1145/3681765.3698455

[15] Yueyang Su, Di Yao, Xiaolei Zhou, Yuxuan Zhang, Yunxia Fan, Lu Bai, and Jingping Bi. 2023. Tripsafe: Retrieving safety-related abnormal trips in real-time with trajectory data. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2446–2450.

[16] 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.

[17] Zheng Zhang, Hossein Amiri, Zhenke Liu, Liang Zhao, and Andreas Züfle. 2024. Large language models for spatial trajectory patterns mining. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection*. Association for Computing Machinery, New York, NY, USA, 52–55.

[18] Zheng Zhang, Hossein Amiri, Dazhou Yu, Yuntong Hu, Liang Zhao, and Andreas Züfle. 2024. Transferable Unsupervised Outlier Detection Framework for Human Semantic Trajectories. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*. Association for Computing Machinery, New York, NY, USA, 350–360.

[19] Yu Zheng, Xing Xie, Quannan Li, and Wei-Ying Ma. 2008. Mining user similarity based on location history. In *SIGSPATIAL'08* (sigspatial'08 ed.). Association for Computing Machinery, New York, NY, USA, Article 34, 10 pages.

[20] Yu Zheng, Xing Xie, and Wei-Ying Ma. 2010. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Eng. Bull.* 33, 2 (2010), 32–39. http://sites.computer.org/debull/A10june/geolife.pdf

[21] Zhenjie Zheng, Zhengli Wang, Liyun Zhu, and Hai Jiang. 2020. Determinants of the congestion caused by a traffic accident in urban road networks. *Accident Analysis & Prevention* 136 (2020), 105327.

[22] 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.

Received 29 August 2025; accepted 25 September 2025