Significant Lagrangian Linear Hotspot Discovery

Yan Li
lixx4266@umn.edu
Dept. of Computer Science & Eng.
University of Minnesota - Twin Cities
Minneapolis, Minnesota

Yiqun Xie
xie@umd.edu
Dept. of Geographical Sciences
University of Maryland
College Park, Maryland

Pengyue Wang
wang6609@umn.edu
Dept. of Mechanical Eng.
University of Minnesota - Twin Cities
Minneapolis, Minnesota

Shashi Shekhar
shekhar@umn.edu
Dept. of Computer Science & Eng.
University of Minnesota - Twin Cities
Minneapolis, Minnesota

William Northrop
wnorthro@umn.edu
Dept. of Mechanical Eng.
University of Minnesota - Twin Cities
Minneapolis, Minnesota

ABSTRACT

Given a collection of multi-attribute trajectories, an event definition, and a spatial network, the Significant Lagrangian Linear Hotspot Discovery (SLLHD) problem finds the paths where records in the trajectories tend to be events in the Lagrangian perspective. The SLLHD problem is of significant societal importance because of its applications in transportation planning, vehicle design, and environmental protection. Its main challenges include the potentially large number of candidate hotspots caused by the tremendous volume of trajectories as well as the non-monotonicity of the statistic measuring event concentration. The related work on the linear hotspot discovery problem is designed in the Eulerian perspective and focuses on point datasets, which ignores the dependence of event occurrence on trajectories and the paths where trajectories are. To solve this problem, we introduce an algorithm in the Lagrangian perspective, as well as five refinements that improve its computational scalability. Two case studies on real-world datasets and experiments on synthetic data show that the proposed approach finds hotspots which are not detectable by existing techniques. Cost analysis and experimental results on synthetic data show that the proposed approach yields substantial computational savings.

CCS CONCEPTS

• **Information systems** → **Geographic information systems**; **Data mining**;

KEYWORDS

hotspot detection, Lagrangian, linear hotspot, multi-attribute trajectories, statistical significance

ACM Reference Format:
Yan Li, Yiqun Xie, Pengyue Wang, Shashi Shekhar, and William Northrop. 2020. Significant Lagrangian Linear Hotspot Discovery. In *13th International Workshop on Computational Transportation Science (IWCTS'20), November 3, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3423457.3429368

IWCTS'20, November 3, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8166-6/20/11…$15.00
https://doi.org/10.1145/3423457.3429368

1 INTRODUCTION

Given a collection of multi-attribute trajectories, an event definition, and a spatial network, the Significant Lagrangian Linear Hotspot Discovery (SLLHD) problem aims to identify the paths where the records in the trajectories tend to be events in the Lagrangian perspective. A multi-attribute trajectory is a sequence of time-stamped records, each of which contains a location and a set of attributes. An event definition is a function mapping from record attributes to a Boolean value. Examples of events include high energy consumption rate and high exhaust emissions value. In the Eulerian perspective, event occurrence depends on the location of records, while in the Lagrangian perspective, trajectories and the paths where trajectories are also affect the probability of event occurring.

The SLLHD problem is critical for applications such as transportation planning, vehicle design, and environmental protection, since moving objects are affected by the paths where they travel. For example, in the Volkswagen emissions scandal, the amount of nitrogen oxides emission from a 2011 Volkswagen Jetta was found to be 37 times over the U.S. limit on up and downhill paths [5]. In the Air France 447 crash, the aircraft's pitot tubes were obstructed by ice crystals, which often happens after an aircraft flies through clouds with small drops [2]. Detecting paths where certain undesirable events in trajectories concentrate can potentially uncover spatial-related causes of the events, which in turn motivates research on ways to prevent (or in the case of desirable events to encourage) the events.

Limitations of related work: SLLHD is a variant of the linear hotspot discovery problem. The most relevant work to SLLHD is the shortest-path (SP) [19] and the all-simple-path (ASP) [22] linear hotspot discovery problems. Both problems, like all traditional hotspot discovery problems (e.g., [11, 15, 24]), focus on hotspots in the Eulerian perspective for individual spatial points and ignore the dependence of event occurrence on trajectories. By contrast, SLLHD detects hotspots of events in trajectories in the Lagrangian perspective.

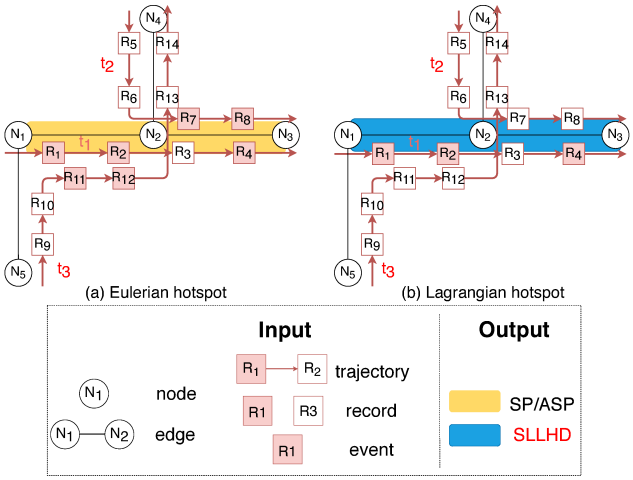

Figure 1: An example of Eulerian and Lagrangian hotspots.

In the Eulerian perspective, hotspots are observed from a specific location in the space [1], and trajectories are ignored. A **Eulerian linear hotspot** is defined as a path where the probability of records being events is higher than that outside the path in the spatial network. It does not consider whether the records are in the trajectories that are on a part or the entire length of the path. For example, Figure 1 shows a part of a spatial network and three trajectories. If the path $[N_1, N_2, N_3]$ is a Eulerian hotspot (Figure 1(a)), the probability of records in t_1, t_2, and t_3 being events should be high on the path. By contrast, the Lagrangian perspective corresponds to an observer moving along a particular path [1]. A **Lagrangian linear hotspot** is defined as a path where in the trajectories that are on the entire length of the path the probability of records being events is higher than that outside the path in the spatial network. If the path $[N_1, N_2, N_3]$ is a Lagrangian hotspot (Figure 1(b)), only the event concentration in t_1 needs to be high, and t_2 and t_3 are ignored since they are not on the entire length of the path.

Since in the Eulerian perspective trajectories are ignored when measuring the probability of records being events, patterns along a path may be overwhelmed by the trajectories that are partially on the path. For example, Figure 2 shows four trajectories (t_1, t_2, t_3, t_4) that log vehicles' energy consumption near an entrance ramp ($[N_5, N_2]$) from a local road ($[N_4, N_5, N_6]$) to a highway (N_1, N_2, N_3). The vehicles traveling freely on the highway, logged by t_2, t_3, t_4, keep almost constant speed, while the vehicle entering the highway from the ramp, logged by t_1, has to merge into the traffic through acceleration. Events refer to high energy consumption. In the Eulerian perspective, on path $[R_7, N_2, R_9]$, which is between the locations of two records R_7 and R_9, there are nine records ($R_7, R_8, R_9, R_{17}, R_{18}, R_{27}, R_{28}, R_{37}, R_{38}$) among which three are events. If in a hotspot the probability of records being events is required to be greater than 0.6, this path is not a hotspot in the Eulerian perspective. Instead, since trajectory t_1 is the only trajectory along the entire length of path $[R_7, N_2, R_9]$, and its records are all events, the path is a hotspot in the Lagrangian perspective. Therefore, in the Eulerian perspective, the concentration of high energy consumption events along

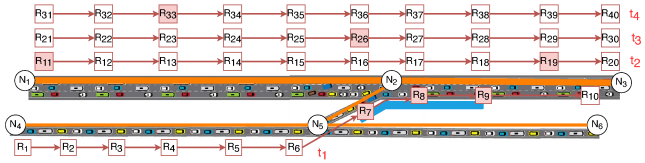

Figure 2: Detecting hotspots in the Eulerian perspective may miss patterns that only exist along a path.

path $[R_7, N_2, R_9]$ for acceleration is overwhelmed by the trajectories partially on the path (i.e., t_2, t_3, t_4). Adopting the Lagrangian perspective distinguishes the experiences on different paths, and helps to highlight the hotspots of the events affected not only by their locations but also by trajectories' paths.

It is common that events in trajectories are affected by trajectories' paths. For example, speed limit varies on roads. Vehicles that just enter a highway from a local road tend to spend more energy than other vehicles on the highway for accelerating and merging into the traffic. In addition, drivers tend to misjudge speed when they exit a highway after long periods of driving at highway speeds, so they are more likely to speed than other drivers on local roads [9]. Brightness is another important heterogeneous factor affecting vehicle status [8]. For example, during winter morning the drivers who just leave a long tunnel tend to brake more frequently since the darkness in the tunnel retards drivers' reaction to bright reflection from snow. Therefore, it is necessary to adopt the Lagrangian perspective to identify hotspots of events in trajectories.

Challenges: SLLHD is challenging due to the potentially large number of candidate paths that can be hotspots given a dataset with millions of trajectories, nodes, and edges in the spatial network. Candidate paths include all the paths between event pairs where there is at least one trajectory. Its amount may be greater than that of simple paths in the network, and it is still increasing every day, since millions of GPS-equipped devices, such as cellphones and vehicle telematics devices, keep reporting their locations. Additionally, the statistic that measures event concentration does not obey the monotonicity property, meaning that there is no ordering between the statistic on a path and its sub-paths, or vice-versa. Furthermore, depending on the method used to determine statistical significance, computation times may also be impacted.

Contributions: Our contributions in this paper are as follows: 1)We formally define the problem of significant Lagrangian linear hotspot discovery (SLLHD); 2) We propose a baseline algorithm to solve the SLLHD problem by enumerating paths between every event pair where there is at least one trajectory, based on which we introduce algorithmic improvements on its scalability; 3) We present two case studies comparing the results of the proposed methods with those of the related work; 4) We conduct experiments to illustrate the computational saving of the proposed algorithm, and to evaluate the detected hotspots quantitatively. To the best of our knowledge, this paper is the first to study hotspot discovery in the multi-attribute trajectories in the Lagrangian perspective.

Workshop relevance: The big data analyzed in this study is from connected vehicles. Depending on the scenarios where the proposed approach is applied, the findings will be beneficial to vehicle design (high energy consumption hotspot), road design

(frequent braking hotspot), transportation planning (low speed hotspot), etc.

The rest of the paper is organized as follows: Section 2 introduces basic concepts and the formal definition of the problem. In Section 3, we propose our approaches for solving the problem. Two groups of controlled experiments are presented in Section 4 to show the advantages of the proposed method over the related work and validate the algorithmic improvements. Two case studies to compare the proposed method and the related work are given in Section 5. In Section 6, we discuss the broad background of the study. Section 7 concludes the paper and presents our future work.

2 PROBLEM FORMULATION

This section introduces basic concepts in the Significant Lagrangian Linear Hotspot Discovery (SLLHD) problem and gives the formal definition of the problem.

2.1 Basic concepts

A **spatial network** $G = (N, E)$ consists of a **node** set N and an **edge** set E, where each element in N is a geo-referenced point, and each element in E is a polyline linking two nodes. For example, in the spatial network shown in Figure 2, there are six nodes (N_1, ..., N_6) and five edges (e.g., $[N_1, N_2]$).

To resolve hotspots to the sub-edge level (i.e., paths between events), dynamic segmentation [19] is conducted, which modifies the input spatial network by forming new nodes at the locations of events, splitting the old edges at the events, and connecting the new nodes through the split edges. We refer to the nodes and edges in the original spatial network as **static nodes and edges**, and the nodes and edges formed by dynamic segmentation as **dynamic nodes and edges**. For simplicity, dynamic nodes are labeled using the events determining their locations. For example, in Figure 2, N_1 is a static node, and $[N_1, N_2]$ is a static edge, and the node formed at R_1 is a dynamic node.

A **path** is an ordered sequence of nodes that are connected by edges, where the origin and the destination are the first and last nodes. For example, in Figure 2 $[N_1, N_2, R_9]$ is a path whose origin and destination are N_1 and R_9 respectively. A **shortest path** is a path that is the shortest according to a measure (typically length) among all the paths that link its origin and destination. A **simple path** is a path that does not repeat nodes.

A **multi-attribute trajectory** is a sequence of time-stamped records. Each record contains a geographic location and a set of attributes. For example, a mobile device trajectory logs the locations where the device was as well as its status (e.g., signal strength, battery state of charge) at certain time points. In Figure 2 a trajectory is represented as a sequence of rectangles linked by arrows (e.g., t_1), and each rectangle is a record (e.g., R_1). **Event**s are records that fulfill certain criteria (e.g., low state of charge). In Figure 2 solid rectangles are the events (e.g., R_7, R_8). A trajectory is on a path if the path links a subset of the records of the trajectory in order. For example, t_1 is on path $[R_1, N_5, N_2]$, but not on path $[N_1, N_2, N_3]$.

A **visited path** of a collection of trajectories is a path where there is at least one trajectory. For example, path $[R_1, N_5, N_2]$ in Figure 2 is a visited path.

To avoid false positive results, a statistical significance test is introduced [14]. The **null hypothesis** (H_0) of the test states that in every trajectory the event concentration is the same throughout the spatial network, while the **alternative hypothesis** (H_1) states that there exist trajectories in which the event concentration inside a hotspot is higher than that outside the hotspot. A **significant hotspot** is defined as a hotspot whose statistical significance p-value is less than or equal to a desired level so that the alternative hypothesis cannot be rejected.

2.2 Problem definition

We formally define the SLLHD problem as follows:

Input:

- A spatial network.
- A collection of multi-attribute trajectories.
- A threshold for the event concentration of a hotspot θ.
- A statistical significance threshold ϕ.

Output: Paths that have event concentration $\geq \theta$ and p-value $\leq \phi$.

Objective:

- Computational efficiency
- Correctness and completeness of results

Constraints:

- A hotspot starts and ends at two dynamic nodes.
- A hotspot is a visited path.
- A hotspot is not a sub-path of any other hotspots.

We assume that only visited paths can be hotspots, since only the trajectories on a path can directly illustrate the difference between the status of objects moving on and off the path in the network.

3 APPROACH

In order to solve the significant Lagrangian linear hotspot discovery (SLLHD) problem, we first introduce the statistic for measuring event concentration. Then we propose an algorithm to detect hotspots, based on which we introduce improvements on its scalability. Last, we describe the method for the statistical significance test using Monte Carlo simulation.

3.1 Statistic for event concentration

We adjust the log-likelihood ratio, which was introduced in SatScan [11], to measure event concentration on a path in the Lagrangian perspective.

We use the Bernoulli model to represent the process of event occurrence in trajectories. Given a trajectory t_i and a path ϕ, let $p_\phi(t_i)$ and $q_\phi(t_i)$ be the probability of a record in t_i being an event on path ϕ and not on path ϕ in the spatial network respectively. Assume that the occurrence of events at each record is independent. The likelihood of path ϕ being a hotspot ($p_\phi(t_i) > q_\phi(t_i)$) is

$$\begin{aligned} L(p_\phi(t_i), q_\phi(t_i)) = {} & p_\phi(t_i)^{n_\phi(t_i)} (1 - p_\phi(t_i))^{\mu_\phi(t_i) - n_\phi(t_i)} \\ & \times q_\phi(t_i)^{n_G(t_i) - n_\phi(t_i)} \\ & \times (1 - q_\phi(t_i))^{(\mu_G(t_i) - \mu_\phi(t_i)) - (n_G(t_i) - n_\phi(t_i))}, \end{aligned}$$

where $n_G(t_i)$ and $\mu_G(t_i)$ are the number of events and records in t_i respectively, while $n_\phi(t_i)$ and $\mu_\phi(t_i)$ are the number of events and records in t_i on path ϕ respectively. If t_i is on ϕ, i.e., $\mu_\phi(t_i) > 0$, conditioned on path ϕ, the maximum likelihood estimate of path ϕ

being a hotspot ($MLE_\phi(t_i)$) is reached when

$$\begin{cases} p_\phi(t_i) = \dfrac{n_\phi(t_i)}{\mu_\phi(t_i)} \text{ and} \\ q_\phi(t_i) = \dfrac{n_G(t_i) - n_\phi(t_i)}{\mu_G(t_i) - \mu_\phi(t_i)}, \end{cases} \quad \text{if } \frac{n_\phi(t_i)}{\mu_\phi(t_i)} > \frac{n_G(t_i) - n_\phi(t_i)}{\mu_G(t_i) - \mu_\phi(t_i)}$$

$$p_\phi(t_i) = q_\phi(t_i) = \frac{n_G(t_i)}{\mu_G(t_i)}, \quad \text{otherwise.}$$

If trajectory t_i is not on path ϕ, i.e., $\mu_\phi(t_i) = n_\phi(t_i) = 0$, $MLE_\phi(t_i)$ is reached when $q_\phi(t_i) = \frac{n_G(t_i)}{\mu_G(t_i)}$. By contrast, the likelihood of path ϕ not being a hotspot ($p_\phi(t_i) = q_\phi(t_i)$) is

$$L_0(t_i) = p_\phi(t_i)^{n_G(t_i)} \times (1 - p_\phi(t_i))^{(\mu_G(t_i) - n_G(t_i)}.$$

So, the maximum likelihood estimate of ϕ not being a hotspot ($MLE_0(t_i)$) is reached when $p_\phi(t_i) = \frac{n_G(t_i)}{\mu_G(t_i)}$, which is the same in cases where t_i is on path ϕ or not. Thus, only when $\frac{n_\phi(t_i)}{\mu_\phi(t_i)} > \frac{n_G(t_i) - n_\phi(t_i)}{\mu_G(t_i) - \mu_\phi(t_i)}$ does the maximum likelihood estimate of ϕ being and not being a hotspot differ according to trajectory t_i.

Assume all trajectories are independent. Given a collection of trajectories T, the likelihood of path ϕ being a hotspot is $L(p_\phi, q_\phi) = \Pi_{t_i \in T} L(p_\phi(t_i), q_\phi(t_i))$, while the likelihood of it not being a hotspot is $L_0 = \Pi_{t_i \in T} L_0(t_i)$. Therefore, we design the statistic for event concentration (*LLR*) to be the difference between the maximum log-likelihood of path ϕ being and not being a hotspot, that is,

$$LLR_\phi(T) = \sum_{t_i \in T} \log(MLE_\phi(t_i)) - \sum_{t_i \in T} \log(MLE_0(t_i)). \quad (1)$$

Because when trajectory t_i is not on path ϕ, $MLE_\phi(t_i) = MLE_0(t_i)$, Equation 1 can be transformed as

$$LLR_\phi(T) = \sum_{t_i \in T_\phi} \log(MLE_\phi(t_i)) - \sum_{t_i \in T_\phi} \log(MLE_0(t_i)), \quad (2)$$

where T_ϕ is the collection of trajectories on path ϕ. In other words, the proposed statistic measures event concentration on a path in the Lagrangian perspective, i.e. it depends only on the trajectories on the path.

It is easy to prove that *LLR* fulfills the following prerequisites of the statistic in the hotspot discovery problem that are listed in [15].

Theorem 1. *(1) Given a fixed number of records, the statistic increases monotonically with the number of events; (2) Given a fixed number of events, the statistic decreases monotonically with the number of records; (3) Given a fixed ratio of events to records, the statistic increases monotonically with the number of records.*

3.2 Baseline algorithm

The idea of the baseline algorithm (SLLHD-Base) to detect hotspots is to traverse through the visited paths between every event pair, select the ones with an *LLR* exceeding the threshold, and remove the ones that are sub-paths of others.

The pseudo-code of the SLLHD-Base algorithm is shown in Algorithm 1. First, the input spatial network is dynamically segmented to include records in trajectories. Then, in Lines 3-9, the algorithm enumerates the visited paths that start from every event using a for loop. The GetVisitedPaths($\hat{G}$, *evt*, T) function (Line 4) yields a collection of visited paths in $\hat{G}$ starting at *evt* using a depth-first search (DFS). The paths with *LLR* exceeding the threshold will be added into the output. The GetLLR(ϕ,T) function (Line 5) traverses through the trajectories (T) to determine whether they are on path ϕ, and calculates the *LLR* of path ϕ using Equation 2. Finally, hotspots which are sub-paths of other hotspots are removed (Line 10).

Algorithm 1 The SLLHD-Base algorithm

Require:
G: A spatial network;
T: Multi-attribute trajectories;
θ: The threshold for the *LLR* of a hotspot.
Ensure: H: Hotspots.
1: $H \leftarrow []$;
2: $\hat{G} \leftarrow$ DynamicSegmentation(G, T);
3: **for all** events *evt* in T **do**
4: **for all** paths ϕ in GetVisitedPaths($\hat{G}$, *evt*, T) **do**
5: **if** ϕ ends at an event and GetLLR(ϕ, T) $\geq \theta$ **then**
6: H.add(ϕ);
7: **end if**
8: **end for**
9: **end for**
10: REMOVE-SUB-PATH(H);

Figure 3 shows a sample input of the problem composed of a spatial network with six nodes and seven edges, three trajectories, and a threshold of *LLR* as 4.0. The algorithm traverses through all events, such as R_1, and enumerates the visited paths from each of them, such as $[R_1, R_2]$, $[R_1, R_{22}]$, $[R_1, R_{13}]$, etc. The *LLR* of $[R_1, R_2]$, $[R_1, R_{22}]$, and $[R_1, R_{13}]$ is 2.98, 4.30, and 4.09 respectively, so $[R_1, R_{22}]$ and $[R_1, R_{13}]$ are hotspots while $[R_1, R_{21}]$ is not. In the last step, $[R_1, R_{22}]$ is removed from the results, because it is a sub-path of $[R_1, R_{13}]$.

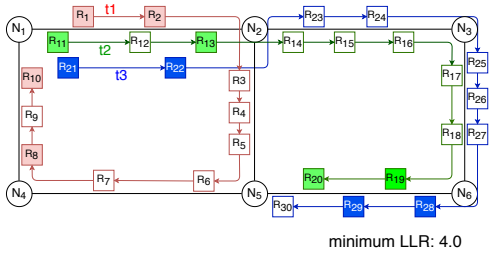

Figure 3: A sample input of the SLLHD problem.

Time complexity analysis: Let $|evt|$ be the total number of events in the trajectories, $|l|$ be the average number of static edges on the path traveled by each trajectory, $|T|$ be the number of trajectories. In the worst case, all trajectories are on different paths, and each event pair is linked by all the visited paths. Since from each event there are at most $|T|$ visited paths, and on each visited path the average number of static/dynamic nodes is at most $|evt|/|T| + |l|$, the time complexity of enumerating the visited paths through DFS is $O(|evt||T|(|evt|/|T| + |l|))$. To compute the *LLR* of a

visited path between two events, trajectories are enumerated. The time complexity of determining whether a trajectory is on a path is $O(|evt|/|T| + |l|)$. Since there are $|evt|^2$ event pairs, and between each pair there are at most $|T|$ visited paths, the time complexity of calculating the *LLR* of all visited paths between event pairs is $O(|evt|^2|T|(|evt|/|T| + |l|)|T|)$. Therefore, the time complexity of the SLLHD-Base algorithm is $O(|evt|^3|T| + |evt|^2|T|^2|l|)$.

3.3 Algorithmic improvements

The two building blocks in the SLLHD-Base algorithm include enumerating visited paths between events, and computing the *LLR* of the paths. To improve computational efficiency, we propose a SLLHD-Scale algorithm that offers five refinements for these building blocks: 1) an edge-based enumeration strategy, 2) an early-stop filter, 3) a bounded-*LLR* filter, 4) a network reduction preprocessing, and 5) a linear scan *LLR* calculation method.

3.3.1 Edge-based enumeration strategy. The idea of edge-based enumeration is to enumerate visited paths first at the static edge level, and then at the sub-edge level. The strategy is based on the following definitions.

A **static path** is a path linking two static nodes. Its first and last static edges are its bounding edges, while the other static edges on it are its bounded edges. A **dynamic path** is a path linking two dynamic nodes formed at events. A dynamic path is a **bounded path** of a static path if its origin and destination are on the two bounding edges of the static path. An **e-edge** is a static edge with events on it. For example, in Figure 3 $[N_1, N_2, N_3, N_6, N_5]$ is a static path. It has a bounded path $[R_2, N_2, N_3, N_6, R_{19}]$, which is a dynamic path. $[N_1, N_2]$ is an e-edge.

The edge-based enumeration strategy has two steps. The first step enumerates all the visited static paths between e-edges through DFS from each e-edge. The second step enumerates the bounded paths of the static paths found in the first step. Take Figure 3 as an example. A DFS for visited static paths between e-edges is conducted from $[N_1, N_2]$, $[N_5, N_6]$, and $[N_1, N_4]$ sequentially. Starting from $[N_1, N_2]$, the algorithm finds visited static paths $[N_1, N_2, N_5, N_4, N_1]$ and $[N_1, N_2, N_3, N_6, N_5]$. To enumerate the bounded paths of $[N_1, N_2, N_5, N_4, N_1]$, the algorithm sets $R_{11}, R_{21}, R_1, R_2, R_{22}$, and R_{13} as the origin, and R_8 and R_{10} as the destination sequentially. $[R_{11}, N_2, N_5, N_4, R_8]$ is an example of a bounded path.

The completeness of the results are maintained because of the following lemma.

Lemma 2. *Every visited dynamic path is a bounded path of a visited static path between two e-edges.*

The proof of lemma 2 is straightforward, since the origin and destination of a dynamic path are two events that must be on e-edges according to the definition of an e-edge.

Based on this strategy, we propose two filters to reduce the number of candidate paths that need to be enumerated and maintain the completeness of the results, according to Theorem 1.

3.3.2 Early-stop filter. The first step of the edge-based enumeration strategy is traversing through the visited static paths between e-edges through DFS. Starting from an e-edge, the algorithm explores visited static paths by adding one static edge at a time to the end of the current path as far as possible before backtracking. The idea of the early-stop filter is to stop the traversal if the largest possible *LLR* of the bounded paths of the visited static paths extended from the current path, denoted as $StopLLR(\phi, T_\phi)$, is smaller than the threshold, where ϕ is the current path and T_ϕ is the collection of trajectories on the current path.

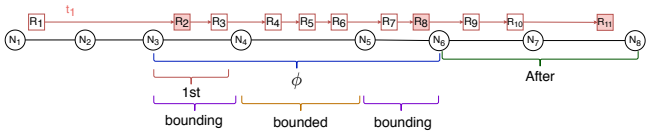

Figure 4: An example of the filters.

In the simplest case where there is only one trajectory t_i on the static path ϕ that is currently explored through DFS, let $1st(\phi)$ be the first edge of path ϕ, and $after(\phi, t_i)$ be the path that trajectory t_i is on after path ϕ. Let $n_{(.)}(t_i)$ and $\mu_{(.)}(t_i)$ be the number of events and records of t_i on specific edges. For example, in Figure 4 trajectory t_1 is on path ϕ. $1st(\phi)$ is edge $[N_3, N_4]$, and $after(\phi, t_1)$ are path $[N_6, N_7, N_8]$. $n_{after(\phi,t_1)}(t_1) = 1$ and $\mu_{1st(\phi)}(t_1) = 1$. We define that

$$StopLLR(\phi, t_i) = LLR_{\hat{\phi}}, \tag{3}$$

such that $\hat{\phi}$ is a bounded path of the visited static path extended from ϕ, and $n_{\hat{\phi}}(t_i) = n_{\phi}(t_i) + n_{after(\phi,t_i)}(t_i)$ and $\mu_{\hat{\phi}} = \mu_{\phi}(t_i) - \mu_{1st(\phi)}(t_i) + n_{1st(\phi)}(t_i) + n_{after(\phi,t_i)}(t_i)$. In other words, the upper bound is reached if there is a bounded path of the visited static paths extended from the current path such that the events on $1st(\phi)$ and $after(\phi, t_i)$ are on it while the records on $1st(\phi)$ and $after(\phi, t_i)$ that are not events are not on it. For example, in Figure 4, the upper bound is reached when the number of events and records on a bounded path of the visited static paths extended from path ϕ are 3 and 8 respectively, so $StopLLR(\phi, t_1) = 0.91$.

We prove the correctness of $StopLLR(\phi, t_i)$ through contradiction. Assume that there is another bounded path $\bar{\phi}$ other than the $\hat{\phi}$ of the visited static paths extended from ϕ such that $LLR_{\hat{\phi}} < LLR_{\bar{\phi}}$. Since both $\bar{\phi}$ and $\hat{\phi}$ must cover all the edges on ϕ except the first one, there are three cases where path $\bar{\phi}$ may differ from path $\hat{\phi}$: 1) all events on $1st(\phi)$ and $after(\phi, t_i)$ are on path $\bar{\phi}$, while some records on $1st(\phi)$ and $after(\phi, t_i)$ that are not events are also on pth $\bar{\phi}$; 2) not all events on $1st(\phi)$ and $after(\phi, t_i)$ are on path $\bar{\phi}$, and records on $1st(\phi)$ and $after(\phi, t_i)$ that are not events are are not on path $\bar{\phi}$; 3) not all events on $1st(\phi)$ and $after(\phi, t_i)$ are on path $\bar{\phi}$, while some records on $1st(\phi)$ and $after(\phi, t_i)$ that are not events are also on path $\bar{\phi}$. In the first two cases, $LLR_{\hat{\phi}} > LLR_{\bar{\phi}}$ due to Theorem 1 (1) and (2). In the third case, if $\mu_{\bar{\phi}}(t_i) < \mu_{\hat{\phi}}(t_i)$, $LLR_{\hat{\phi}} > LLR_{\bar{\phi}}$ due to Theorem 1 (2) and (3); if $\mu_{\bar{\phi}}(t_i) \geq \mu_{\hat{\phi}}(t_i)$, $LLR_{\hat{\phi}} > LLR_{\bar{\phi}}$ due to Theorem 1 (1) and (2), since the ratio of events to records on $\hat{\phi}$ is greater than that on $\bar{\phi}$. Thus, in all the three cases, $LLR_{\hat{\phi}} > LLR_{\bar{\phi}}$, which is in contradiction with the assumption.

Suppose there is a collection of trajectories T_ϕ on the current path ϕ. The upper bound is

$$StopLLR(\phi, T_\phi) = \Sigma_{t_i \in T_\phi} StopLLR(\phi, t_i). \tag{4}$$

In other words, the upper bound is reached if the trajectories in T_ϕ are on the same path after ϕ, and on the path the *LLR* given every

trajectories reaches its upper bound. The proof is straightforward since $StopLLR$ and LLR are always positive. If $StopLLR(\phi, T_\phi)$ is smaller than the threshold, further exploration by extending ϕ is not necessary.

3.3.3 Bounded LLR filter. The second step of the edge-based enumeration strategy is enumerating the bounded paths of visited static paths. The idea of the bounded LLR filter is that if the upper bound of the LLR, denoted as $BoundedLLR(\phi, T)$, of the bounded paths of a visited static path ϕ is smaller than the threshold, the enumeration can be terminated.

Again, we start from the simplest case, where there is only one trajectory t_i on a visited static path ϕ. Let $bounding(\phi)$ and $bounded(\phi)$ be the bounding and bounded edges of ϕ respectively. For example, in Figure 4, ϕ is $[N_3, N_4, N_5, N_6]$, and $bounding(\phi)$ includes edge $[N_3, N_4]$ and edge $[N_5, N_6]$, and $bounded(\phi)$ includes edge $[N_4, N_5]$. In this case, we define that

$$BoundedLLR(\phi, t_i) = LLR_{\hat{\phi}}, \tag{5}$$

such that $\hat{\phi}$ is a bounded path of ϕ, and $n_{\hat{\phi}}(t_i) = n_{bounding(\phi)}(t_i) + n_{bounded(\phi)}(t_i)$ and $\mu_{\hat{\phi}} = n_{bounding(\phi)}(t_i) + \mu_{bounded(\phi)}(t_i)$. In other words, the upper bound is reached if the events on $bounding(\phi)$ are on the bounded path while the records on $bounding(\phi)$ that are not events are not on the path. For example, in Figure 4, the upper bound is reached when the number of events and records on a bounded path of ϕ are 2 and 5 respectively, so $BoundedLLR(\phi, t_1) = 0.58$.

The proof of $BoundedLLR(\phi, t_i)$ is similar to that of $StopLLR(\cdot)$ according to Theorem 1.

Suppose there is a collection of trajectories T_ϕ on a visited static path ϕ. The upper bound is

$$BoundedLLR(\phi, T_\phi) = \Sigma_{t_i \in T_\phi} BoundedLLR(\phi, t_i). \tag{6}$$

In other words, the upper bound is reached if on a bounded path of ϕ the LLR given every trajectory reaches its upper bound. This upper bound is valid since $BoundedLLR$ and LLR are always positive. If $BoundedLLR(\phi, T_\phi)$ is lower than the threshold, we do not need to traverse through the bounded paths of ϕ.

3.3.4 Network reduction preprocessing. Our next to last refinement reduces the network size. In the first step of the edge-based enumeration strategy described earlier, the algorithm explores the visited static paths between e-edges through DFS by adding one static edge at a time to the end of the path that is currently being explored. However, in this paper we focus only on paths between events. By pre-computing the visited paths without events on them, we can reduce the enumeration needed to explore them repeatedly. Thus, we propose the following network reduction preprocessing.

Given a spatial network, a collection of trajectories with events on them, an **e-edge network** is a spatial network composed of: 1) nodes, each of which represents an e-edge; and 2) edges, each of which represents a visited path with no event on it. Two nodes in an e-edge network are linked if the two e-edges represented by the two nodes are connected by a visited path with no event on it in the spatial network. For example, given the spatial network and trajectories in Figure 3, we can get the e-edge network shown in Figure 5. There are three e-edges, namely, edges $[N_1, N_2]$, $[N_4, N_1]$, and $[N_6, N_5]$. The e-edges are connected by two visited static paths, namely, paths $[N_2, N_5, N_4]$ and $[N_2, N_3, N_6]$.

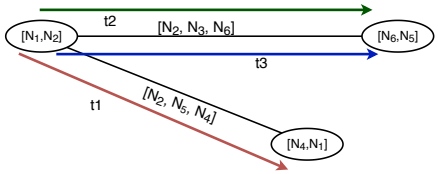

Figure 5: The e-event network generated from the spatial network and the trajectories in Figure 3.

Once an e-event network is constructed, it can be used to enumerate all the static paths whose bounded path may be hotspots.

3.3.5 Linear scan LLR calculation. Our final refinement reduces the computational cost of LLR calculation. The baseline algorithm computes the LLR of every dynamic path by first determining the trajectories on the path and then calculating LLR using Equation 2. However, there exists redundant computation because of the following theorem.

Theorem 3. *If path ϕ_1 is a sub-path of a path ϕ_2, that is, all the nodes of ϕ_1 are connected by ϕ_2 in order without any other nodes in between, the trajectories on ϕ_2 is a subset of the trajectories on ϕ_1.*

Therefore, if we know the trajectories on a path, then querying the trajectories on the paths extended from the current path through DFS needs researching only the trajectories on the current path. It is not necessary to search the entire trajectory dataset. In addition, by keeping records of the variables needed to calculate LLR and the upper bounds for the two filters during DFS, we can avoid counting the number of records and events repeatedly. The variables that have to be saved include the trajectories on the current path, the number of records and events on the current path, the number of records and events on the first static edge of the current path, and the number of records and events after the current path.

3.3.6 Time complexity analysis. Let $|evt|$ be the number of all events in the trajectories, $|l|$ be the average number of static edges in the path traveled by each trajectory, and $|T|$ be the number of trajectories. In the worst case, all trajectories are on different paths, and each event pair is linked by all the visited paths. When constructing the e-edge network, every path the trajectories are on has to be enumerated once, so the time complexity is $O(|l||T|)$. Let $|e\_edge|$ be the number of all e-edges. Without the early-stop filter and the bounded-LLR filter, the time complexity of enumerating the visit static paths between e-edges is $O(|e\_edge|^2|T|)$. Once these paths have been enumerated, each of their bounded paths is enumerated once, giving a total time complexity of enumerating the visited dynamic paths of $O((|e\_edge|^2|T| + |evt|^2)\alpha\beta)$, where α and β are the percentage of the visit static paths remained after applying the two filters. Because of the linear scan LLR calculation, the query of whether a trajectory is on a static edge would be conducted $|e\_edge| + |l|$ times for each trajectory. Therefore, the time complexity of the SLLHD-Scale algorithm is $O((|evt|^2 + |e\_edge||T| + |l||T|)\alpha\beta)$, which is much lower than that of the SLLHD-Base algorithm, $O(|evt|^3|T| + |evt|^2|T|^2|l|)$.

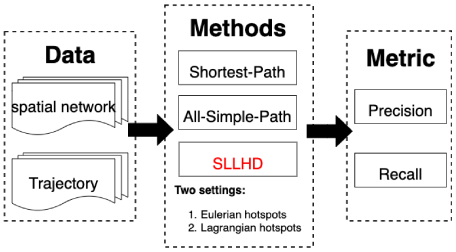

Figure 6: Result quality experiment design.

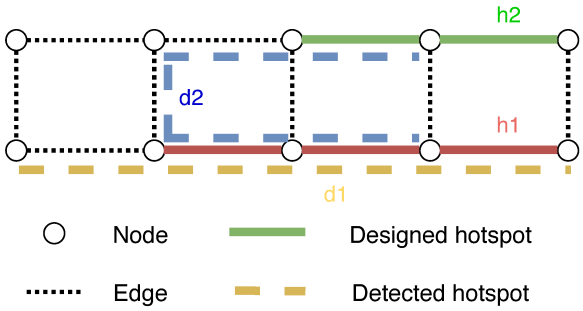

Figure 7: An example of maximal intersection over union.

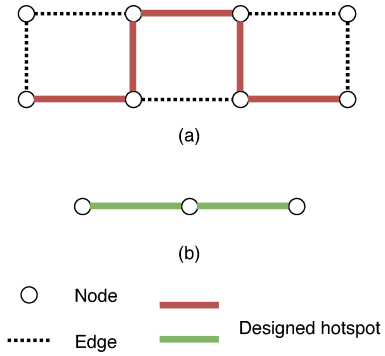

Figure 8: Two designed hotspots.

3.4 Statistical significance test

Each hotspot is evaluated for statistical significance using Monte Carlo simulation, which is a randomization test to get the distribution of *LLR*. We run the Monte Carlo simulation m times to get the distribution of the greatest *LLR* in each iteration under the null hypothesis. In each iteration, we use Mantel's permutational approach [12] to generate the simulated data. That is, the spatial locations of the records in trajectories and the ratio of events to records do not change, while the positions of the events are shuffled randomly, forming a new collection of trajectories T_R. Then, we detect the hotspot with the greatest *LLR* in T_R. Once we get the distribution, the statistical significance p-value of each hotspot is determined by the friction of the greatest *LLR* that is greater than its *LLR*.

4 EXPERIMENTS

We conducted two sets of experiments to: 1) compare the result quality of the hotspots detected by the proposed approach and the related work; and 2) compare the computational performance of the proposed SLLHD-Base and SLLHD-Scale algorithms.

4.1 Result quality

We designed the experiments as shown in Figure 6 to compare the result quality of the hotspots detected by the proposed approach (SLLHD) and the related work in two settings. In the first setting, all hotspots were Eulerian hotspots, while in the other setting, all hotspots were Lagrangian hotspots. The related work compared included the shortest-path (SP) method [19] and the all-simple-path (ASP) method [22]. The evaluation metrics were precision and recall. We set that if the maximal intersection over union (MaxIoU) of a detected hotspot with designed hotspots exceeded 0.6, the detect hotspot was valid. For example, in Figure 7 there is a spatial network composed of ten nodes and 13 edges, two designed hotspots (h1 and h2), and two detected hotspots (d1 and d2). Suppose the edges share the same length 1. The lengths of the intersections of h1 and h2 with d1 are 3 and 0, and the lengths of the unions of h1 and h2 with d1 are 4 and 6, so the MaxIoU of d1 is 3/4. This value exceeds the threshold, so d1 is a valid hotspot. By contrast, the lengths of the intersections of h1 and h2 with d2 are 2 and 1, and the lengths of the unions of h1 and h2 with d2 are 6 and 6, so the MaxIoU of d2 is 2/6 and d2 is not valid.

We conducted the experiments on synthetic data. Limited by the computational complexity of the ASP method, the volume of the synthetic data is not large. The spatial network was a 7 by 7 grid with 49 nodes and 84 edges. The edges shared the same length. We designed hotspots with two shapes shown in Figure 8. One hotspot (Figure 8(a)) consisted of 5 edges not along a shortest path so that it should be ignored by the SP method, which focuses on hotspots along shortest paths. The second hotspot (Figure 8(b)) consisted of 2 edges so that it was within the set of candidate hotspots of all three methods. In each run of the experiment, the designed hotspots were randomly positioned in the spatial network, and 50 trajectories that traveled along 7 edges were generated. Five trajectories were positioned along each of the two hotspots (Figure 8), and 40 other trajectories were partially on the two hotspots. The number of trajectory records on an edge was set as 2. We conducted experiments in the two settings 100 times to highlight the advantages of the proposed method.

Setting 1: Both designed hotspots were Eulerian linear hotspots. The probability of records being events in the hotspots was 0.8, while that outside the hotspots was 0.2.

Setting 2: Both designed hotspots were Lagrangian linear hotspots. In the 10 trajectories that were along the entire length of the hotspots, the probability of records being events in the hotspots was 0.8, while that outside the hotspots was 0.2. In the other 40 trajectories, the probability of records being events was 0.2 throughout the network.

The results of the experiments in the two settings are shown in Table 1. In setting 1, the result quality of the ASP method and the proposed method was much higher than that of the SP method, since one of the designed hotspots was completely ignored by the

Table 1: Quality results for the three methods.

	Setting 1		Setting 2	
	Precision	Recall	Precision	Recall
SP	0.43	0.445	0.25	0.20
ASP	0.955	0.97	0.37	0.24
SLLHD	0.945	0.96	0.95	0.94

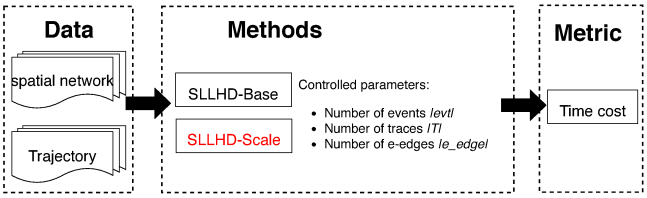

Figure 9: Computational performance experiment design.

SP method. The result quality of the proposed method and the ASP method was similar. In setting 2, the result quality of both the SP and the ASP methods decreased dramatically. A potential cause to the results was that the 40 trajectories that were partially located on the designed hotspots decreased the event concentration calculated by the ASP method, since the ASP method was designed for individual points and ignored trajectories. Therefore, the experiments on result quality indicate that the proposed method is able to detect Eulerian hotspots that can be detected by the ASP method, as well as Lagrangian hotspots that are ignored by the state-of-the-art related work.

4.2 Computational performance

To compare the computational performance of the proposed algorithms (SLLHD-Base and SLLHD-Scale), we designed the experiments as shown in Figure 9, where the controlled parameters were the number of events in the trajectories $|evt|$, the number of trajectories $|T|$, and the number of e-edges $|e\_edge|$. For each experiment, we executed the algorithms 10 times and compared the average execution time of each algorithm.

The experiments were conducted on synthetic data. The spatial network was a 50 by 50 grid with 2500 nodes and 4900 edges. The trajectories were generated from a random origin and through the random walk with equal transition probability between nodes. Each trajectory traveled along 25 edges, and its records on each trajectory were uniformly distributed. E-edges were randomly selected from edges with trajectories on them. The ratio of events to records on an e-edge was a random value generated from a uniform distribution between 0 and 1.

All experiments were performed on a single server with a quad-core Intel(R) Xeon(R) CPU E5-2623 v3 (3.00GHz) and 64GB memory. All algorithms were implemented in Java, and the version of the Java runtime was 11.0.3.

The experiments were designed to answer the following questions: 1) How do the proposed algorithms compare in efficiency? 2) How are the algorithms affected by the total number of events, the number of trajectories, and the number of e-edges?

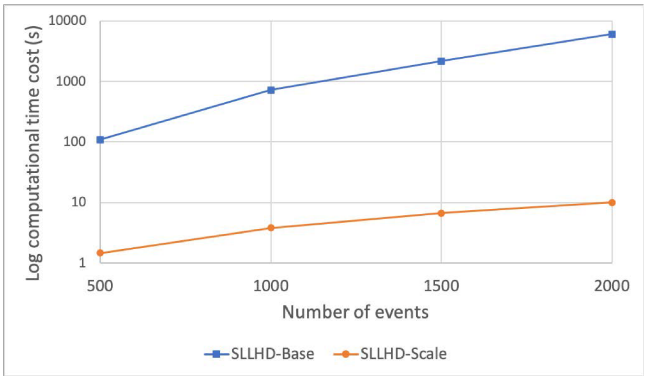

Figure 10: The computational performance with varying number of events.

Figure 11: The computational performance with varying number of trajectories.

4.2.1 Number of events. In this set of experiments, we set the number of trajectories to 100, the number of records to 10000, and the number of e-edges to 500, and varied the number of events from 500 to 2000. In all cases SLLHD-Scale executed the fastest, which accords with the time complexity analysis (Figure 10). Additionally, its time cost increased more slowly than SLLHD-Base as the number of events increased, which is also consistent with the time complexity of SLLHD-Scale and SLLHD-Base.

4.2.2 Number of trajectories. Here, we set the number of records, events and e-edges to 10000, 1000, and 500 respectively, but varied the number of trajectories from 50 to 200. The results are shown in Figure 11. Again, the computational time savings of SLLHD-Scale were clear. In addition, its time cost increased more slowly compared to the baseline algorithm.

4.2.3 Number of e-edges. In these experiments, we set the number of trajectories, records, and events to 100, 10000, and 1000 respectively, but varied the number of e-edges from 250 to 1000. As we can see, the time costs for the SLLHD-Scale algorithm remained much lower than that of the SLLHD-Base algorithm. In addition, the number of event edges did not affect the baseline algorithm

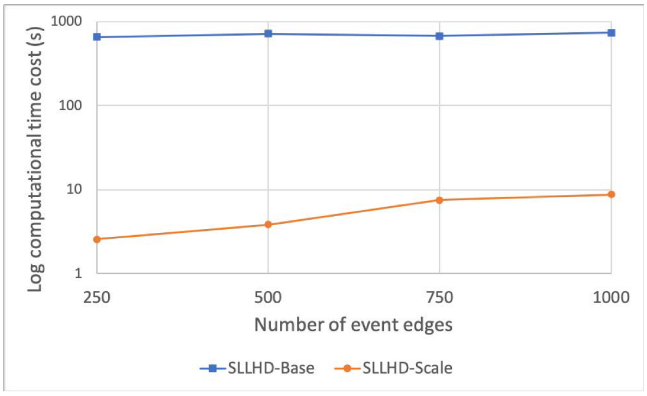

Figure 12: The computational performance with varying number of e-edges.

much, but it was positively correlated with the time cost of the SLLHD-Scale algorithm.

5 CASE STUDY

We compared our approach against the shortest-path (SP) method [19] in two case studies using real-world datasets. The all-simple-path method introduced in [22] is too time-consuming to be applied in these two case studies. Its time cost in a spatial network containing 2000 nodes is about 28 hours [22]. In order to control the effect of the statistic for event concentration on the results, we used the *LLR* proposed in this paper in both our approach and the SP method.

5.1 Case 1: Metro Transit - Twin Cities, MN

The first case study was conducted on a dataset from a Metro Transit bus in Minneapolis-St. Paul, MN [10]. The data contained 212 trajectories and 1 million records in total. The records with the top 1% fuel use rate in each trajectory were labeled as high energy consumption events. The study area was the road network in the minimum orthogonal bounding rectangle of the trajectories, containing 90285 road segments and 62103 road intersections. The minimum *LLR* of a hotspot was 10, the statistical significance threshold was 0.01, and the number of Monte Carlo simulation was 1000.

The results are shown in Figure 13, where records and events in the trajectories are blue and red dots, and the four hotspots detected by the proposed method are highlighted in yellow and green. The SP method detected the three yellow hotspots, namely hotspots 2, 3, and 4, but missed the green one (i.e., hotspot 1), since it is not along a shortest path. There are 238 events and 1689 points in hotspot 1, whose LLR is 390.96 and statistical significance is 0.001, making it unlikely to be a false positive result.

5.2 Case 2: Snowplow - MN

The second case study was conducted on a dataset from a municipal snowplow in Minnesota. The data contained 980 trajectories and 60 million records in total. The records with the top 1% fuel use rate are labeled as high energy consumption events. The study area was

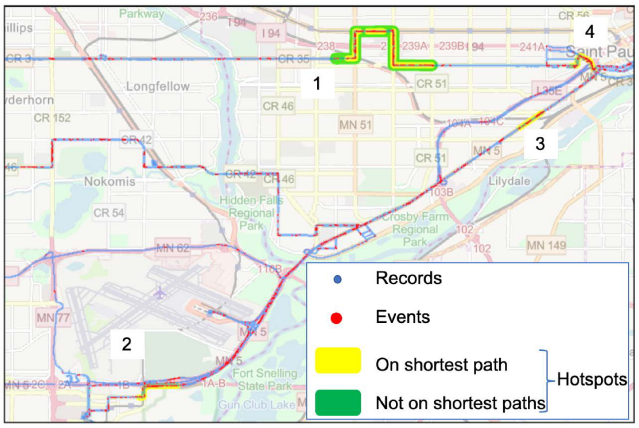

Figure 13: Results of case study 1.

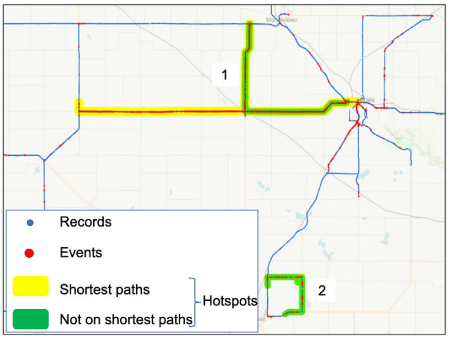

Figure 14: Results of case study 2.

the road network in the minimum orthogonal bounding rectangle of the trajectories, containing 18390 road segments and 12737 road intersections. The minimum *LLR* of a hotspot was 10, the statistical significance threshold was 0.01, and the number of Monte Carlo simulation was 1000.

As shown in Figure 14, similar to the results in the first case study, the hotspots detected by the proposed method are highlighted in yellow and green. The yellow hotspots are also detected by the SP method, while the green ones (i.e., hotspots 1 and 2) are not. The LLR of the hotspots 1 and 2 is 32.00 and 25.35, and their statistical significance is 0.001 and 0.004 respectively, so neither of the results is likely to be a false positive. Therefore, both case studies show that the proposed approach can detect hotspots that are not detectable by the related work.

6 DISCUSSION

Spatial hotspot discovery has been widely studied in the last decade because of its importance in application areas such as public health and criminology. These research generally falls into two groups, i.e., methods based on spatial autocorrelation analysis and those based on spatial scan statistics. Spatial autocorrelation analysis based methods [6, 23] mainly apply spatial statistics such as Moran's I [13], Getis-Ord Gi* [16] to identify hotspots in pre-defined regions. Instead, spatial scan statistics based methods query all regions that

meet a certain criterion in the study area. Based on the spatial footprint of the hotspot, the spatial scan statistics based methods are in two groups, i.e., Euclidean-based (e.g., circles [11], rectangles [15], ellipses [20], rings [4], density-based shapes [25]) and network-based (e.g., linear hotspot [18, 19, 22], subgraph [3, 17, 21]). Since these methods are designed for individual point data but ignore trajectories, they will miss some interesting hotspots that can be detected by our proposed method.

Also extensively studied in the last decade is pattern mining in trajectories, such as moving together patterns (e.g., flocks, convoys, swarms, traveling companions, and gatherings) and trajectory clustering [7, 26]. However, these studies focus more on the concentration of trajectories, not on the concentration of particular events in multi-attribute trajectories. Therefore, they are not applicable to this problem.

7 CONCLUSION AND FUTURE WORK

In this paper, we studied the problem of finding paths where events in multi-attribute trajectories concentrate in a Lagrangian perspective. After formally defining the problem, we proposed a baseline algorithm and five refinements that improved its scalability while maintaining correctness and completeness. We conducted two case studies using Twin-Cities Metro Transit data and Minnesota snowplow data that show the proposed approach finds hotspots which are not detectable by the state-of-the-art techniques. We also conducted experiments on synthetic data to illustrate that the proposed method was able to detect hotspots that were neglected by the related work, and that the refinements yielded substantial computational time savings.

In the future, we plan to explore significant Lagrangian linear hotspot discovery with continuous feature values, as well as the influence of different sampling rates of trajectories.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grants No. 1737633, 1901099, 1916518, and IIS-1218168, the USDOD under Grants No. HM1582-08-1-0017 and HM0210-13-1-0005, the Advanced Research Projects Agency-Energy, U.S. Department of Energy under Award No. DE-AR0000795, the NIH under Grant No. UL1 TR002494, KL2 TR002492, and TL1 TR002493, the USDA under Grant No. 2017-51181-27222. The authors would like to thank Kim Koffolt and the University of Minnesota Spatial Computing Research Group for their comments.

REFERENCES

[1] G.K. Batchelor. 2000. *An Introduction to Fluid Dynamics*. Cambridge University Press.

[2] Nicola Clark. 2012. Report on Air France Flight 447 Cites Conflicting Data in Cockpit. *The New York Times* (July 2012). https://www.nytimes.com/2012/07/06/world/europe/air-france-flight-447-report-cites-confusion-in-cockpit.html

[3] Marcelo Azevedo Costa, Renato Martins AssunÃ§Ã£o, and Martin Kulldorff. 2012. Constrained spanning tree algorithms for irregularly-shaped spatial clustering. *Computational Statistics & Data Analysis* 56, 6 (June 2012), 1771–1783. https://doi.org/10.1016/j.csda.2011.11.001

[4] E. Eftelioglu, S. Shekhar, J. M. Kang, and C. C. Farah. 2016. Ring-Shaped Hotspot Detection. *IEEE Transactions on Knowledge and Data Engineering* 28, 12 (Dec. 2016), 3367–3381. https://doi.org/10.1109/TKDE.2016.2607202

[5] Guilbert Gates, Jack Ewing, Karl Russell, and Derek Watkins. 2015. How Volkswagen's 'Defeat Devices' Worked. *The New York Times* (Oct. 2015). https://www.nytimes.com/interactive/2015/business/international/vw-diesel-emissions-scandal-explained.html

[6] Mohammadhossein Ghahramani, MengChu Zhou, and Chi Tin Hon. 2018. Mobile phone data analysis: A spatial exploration toward hotspot detection. *IEEE Transactions on Automation Science and Engineering* 16, 1 (2018), 351–362.

[7] Joachim Gudmundsson, Marc van Kreveld, and Frank Staals. 2013. Algorithms for hotspot computation on trajectory data. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 134–143.

[8] Yueqi Hu, Haoxue Liu, and Tong Zhu. 2019. Influence of spatial visual conditions in tunnel on driver behavior: Considering the route familiarity of drivers:. *Advances in Mechanical Engineering* (May 2019). https://doi.org/10.1177/1687814019853661

[9] Richard J Jagacinski and John M Flach. 2018. *Control theory for humans: Quantitative approaches to modeling performance*. CRC Press.

[10] Andrew J. Kotz and William F. Northrop. [n.d.]. Metro Transit Diesel Bus Engine Measurement Data for 19 Days in Winter 2014 in Minneapolis-St. Paul, MN, USA. type: dataset.

[11] Martin Kulldorff. 1997. A spatial scan statistic. *Communications in Statistics - Theory and Methods* 26, 6 (Jan. 1997), 1481–1496. https://doi.org/10.1080/03610929708831995

[12] Nathan Mantel. 1967. The Detection of Disease Clustering and a Generalized Regression Approach. *Cancer Research* 27, 2 Part 1 (Feb. 1967), 209–220.

[13] Patrick AP Moran. 1950. Notes on continuous stochastic phenomena. *Biometrika* 37, 1/2 (1950), 17–23.

[14] Jerome L Myers, Arnold Well, and Robert Frederick Lorch. 2010. *Research design and statistical analysis*. Routledge.

[15] Daniel B. Neill and Andrew W. Moore. 2004. Rapid Detection of Significant Spatial Clusters. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*. ACM, New York, NY, USA, 256–265. https://doi.org/10.1145/1014052.1014082

[16] J Keith Ord and Arthur Getis. 1995. Local spatial autocorrelation statistics: distributional issues and an application. *Geographical analysis* 27, 4 (1995), 286–306.

[17] Benjamin Romano and Zhe Jiang. 2017. Visualizing traffic accident hotspots based on spatial-temporal network kernel density estimation. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.

[18] L. Shi and V. P. Janeja. 2011. Anomalous Window Discovery for Linear Intersecting Paths. *IEEE Transactions on Knowledge and Data Engineering* 23, 12 (Dec. 2011), 1857–1871. https://doi.org/10.1109/TKDE.2010.212

[19] X. Tang, E. Eftelioglu, D. Oliver, and S. Shekhar. 2017. Significant Linear Hotspot Discovery. *IEEE Transactions on Big Data* 3, 2 (June 2017), 140–153. https://doi.org/10.1109/TBDATA.2016.2631518

[20] Xun Tang, Emre Eftelioglu, and Shashi Shekhar. 2015. Elliptical Hotspot Detection: A Summary of Results. In *Proceedings of the 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data (BigSpatial'15)*. ACM, New York, NY, USA, 15–24. https://doi.org/10.1145/2835185.2835192

[21] Xun Tang, Emre Eftelioglu, and Shashi Shekhar. 2017. Detecting Isodistance Hotspots on Spatial Networks: A Summary of Results. In *Advances in Spatial and Temporal Databases*. Springer International Publishing, 281–299.

[22] Xun Tang, Jayant Gupta, and Shashi Shekhar. 2019. Linear Hotspot Discovery on All Simple Paths: A Summary of Results. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19)*. Association for Computing Machinery, Chicago, IL, USA, 476–479. https://doi.org/10.1145/3347146.3359100

[23] Tao Wang, Fuzhong Xue, Yongjin Chen, Yunbo Ma, and Yanxun Liu. 2012. The spatial epidemiology of tuberculosis in Linyi City, China, 2005–2010. *BMC public health* 12, 1 (2012), 885.

[24] Yiqun Xie and Shashi Shekhar. 2019. A Nondeterministic Normalization based Scan Statistic (NN-scan) towards Robust Hotspot Detection: A Summary of Results. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 82–90. https://doi.org/10.1137/1.9781611975673.10

[25] Yiqun Xie and Shashi Shekhar. 2019. Significant DBSCAN towards Statistically Robust Clustering. In *Proceedings of the 16th International Symposium on Spatial and Temporal Databases*. ACM, 31–40.

[26] Yu Zheng. 2015. Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.* 6, 3 (May 2015), 29:1–29:41. https://doi.org/10.1145/2743025