A Unified Framework for Robust and Efficient Hotspot Detection in Smart Cities

YIQUN XIE, University of Minnesota – Twin Cities
SHASHI SHEKHAR, University of Minnesota – Twin Cities

Given N geo-located point instances (e.g., crime or disease cases) in a spatial domain, we aim to detect sub-regions (i.e., hotspots) that have a higher probability density of generating such instances than the others. Hotspot detection has been widely used in a variety of important urban applications, including public safety, public health, urban planning, equity, etc. The problem is challenging because its societal applications often have low-tolerance for false positives, and require significance testing which is computationally intensive. In related work, the spatial scan statistic introduced a likelihood ratio based framework for hotspot evaluation and significance testing. However, it fails to consider the effect of spatial nondeterminism, causing many missing detections. Our previous work introduced a nondeterministic normalization based scan statistic to mitigate this issue. However, its robustness against false positives is not stably controlled. To address these limitations, we propose a unified framework which can improve the completeness of results without incurring more false positives. We also propose a reduction algorithm to improve the computational efficiency. Experiment results confirm that the unified framework can greatly improve the recall of hotspot detection without increasing the number of false positives, and the reduction algorithm can greatly reduce execution time.

CCS Concepts: • **Information systems → Clustering**; **Geographic information systems**; **Data analytics**.

Additional Key Words and Phrases: unified framework, hotspot, smart cities

ACM Reference Format:
Yiqun Xie and Shashi Shekhar. 2018. A Unified Framework for Robust and Efficient Hotspot Detection in Smart Cities. *J. ACM* 37, 4, Article 111 (August 2018), 28 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Given a collection of N geo-located point instances (e.g., crime or disease cases, locations of service centers or facilities) in a spatial domain, the problem of hotspot detection aims to find sub-regions that have a higher probability density of generating or having such instances compared to the rest of the space.

Hotspot detection is a critical topic in spatial data mining [4, 37, 38, 46], and has wide and important applications in smart cities and communities (e.g., [1, 37, 47]), including public safety, public health, urban planning, equity, economics, etc. In public safety, regions with significantly high concentration of crime activities can signal the need for administrative inspection and intervention [23, 26]. Such hotspots can also help police officers locate the places of residence of serial criminals (e.g., arsonists) [8]. In public health, hotspots of infections indicate outbreaks of disease [18, 20,

Authors' addresses: Yiqun Xie, xiexx347@umn.edu, University of Minnesota – Twin Cities, Department of Computer Science and Engineering; Shashi Shekhar, shekhar@umn.edu, University of Minnesota – Twin Cities, Department of Computer Science and Engineering.

0004-5411/2018/8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

36, 47], which can help officials allocate medical resources and research efforts. In environmental science, hotspots of pollution (e.g., E. coli contamination) help researchers identify the cause and improve urban water quality [16, 17]. In urban resilience studies, hotspots of building permits can be used to evaluate the status of recovery from natural disasters (e.g., hurricanes) [40]. In transportation, roads with significantly high concentration of pedestrian fatalities indicate unsafe conditions of related infrastructures (e.g., side walk) [41]. These are just a few of many applications of hotspot detection.

The problem is challenging due to the high cost of false positives in urban and smart-city applications (e.g., [1, 18, 37, 46, 48]). For example, falsely identifying a neighborhood as a hotspot of robbery may cause unnecessary social anxiety, lower property values and hurt small businesses. As a result, statistical significance is often required. In hotspot detection, the statistical distribution of test statistics (i.e., scores of hotspot candidates) is often unknown and cannot be computed in closed-form [18, 31]. Thus, the p-values of detections require a large number of Monte Carlo trials to estimate, making the problem computationally intensive.

In related work, the spatial scan statistic [18] is the most widely used approach for hotspot detection, and its corresponding software, SaTScan [3], has received over 10,000 downloads. The spatial scan statistic provides a likelihood ratio based framework for hotspot evaluation and significance testing. By default, it detects hotspots that have a circular shape. In recent years, most efforts in the data mining community have focused on extending this framework to detect hotspots of other geometric shapes (e.g., ring [7, 8], linear [35, 39], rectangle [31, 32], ellipse [21]) to meet specific domain needs. Meanwhile, many other efforts have also been made to extend the original spatial scan statistic to detect multi-variate hotspots [22, 28], spatiotemporal (i.e., emerging) hotspots [27, 29] and continuous value (e.g., lifespan) based hotspots [11, 19]. Studies have also tried to analyze and mitigate the uncertainty (e.g., geographically aggregated data) and inaccuracy issues in hotspot detection [24, 25]. While various extensions exist, they either directly adopted the likelihood ratio framework provided by the spatial scan statistic (e.g., extensions for new shapes [7, 8, 21, 32, 39]) or they enriched the likelihood ratio specifically for a certain application scenario (e.g., multivariate, temporal or continuous value based hotspots [19, 22, 27, 29]). In both cases, the methods do not attempt to change the fundamental way that the likelihood ratio is formulated or defined in the spatial scan statistic. [1]

As our previous study [48] showed, however, the likelihood ratio definition in the spatial scan statistic fails to consider the effect of spatial nondeterminism (details in Sec. 3.2). This introduces bias and causes the likelihood ratio values of many false positives to be higher than those of the true hotspots, leading to missing detections in the output. To mitigate this issue, our previous work [48] proposed a nondeterministic normalization based scan statistic (NN-scan, Sec. 3) to improve the completeness of the results. However, in order to model the spatial nondeterminism and make it computationally feasible, NN-scan requires a pre-defined set of hotspot sizes (e.g., radii in the case of circular hotspots). While a greater number of candidate sizes can make the enumeration space more complete, through new exploration, we find that NN-scan's robustness against false positives decreases as the number of candidate sizes increase (details in Sec. 3.3).

To address these limitations, we propose a unified framework that integrates NN-scan and the likelihood ratio based framework. This unified framework has two major advantages: (1) it removes the need for a pre-defined set of hotspot sizes; (2) it improves the completeness of results without any loss of robustness against false positives. We further propose a reduction algorithm to improve the computational efficiency of the new approach. Table 1 shows the novelty of the proposed work.

[1]Since the goals of these major extensions are different from ours, we provide a detailed discussion of these extensions in Sec. 7 instead of here to avoid diluting the focus.

Table 1. Novelty of the proposed work

Criteria	Spatial scan statistic	NN-scan (conf.)	Unified (proposed)
Avoid missing detections caused by the bias of a test statistic	No	Yes	Yes
Stable enforcement of the significance level	Yes	No	Yes

Through detailed analytical validation and experiments, we confirm that the unified framework can greatly improve the recall on hotspot detection without incurring more false positives, and the reduction algorithm outperforms the baseline algorithm by orders of magnitudes.

1.1 Scope and Outline

Since this paper is an extension of our previous study [48], the scope and the problem definition remain the same. Specifically, we will use circular shaped candidate regions since circles are still the most used shape in both research and applications of hotspot detection. In addition, the discussion will focus on the continuous version of hotspot detection, where statistical population is proportional to geometric area. This continuous version is also widely used in hotspot detection (e.g., [8, 16, 17, 23, 26, 40, 41, 45]). Algorithm modifications for the discrete version, in which a statistical population is available and given as a separate point distribution, are beyond the scope of this specific extension and will be explored in future work.

In this work, we propose the following main extensions to our previous study [48]: (1) We provide a new analysis of the theoretical limitations of the likelihood-ratio based scan statistics, and we identify a new instability issue in our previous NN-scan method (i.e., instability in robustness against false positives); (2) We analyze the high-level structures of existing methods and propose a new unified framework that combines the advantages of the methods while avoiding their disadvantages; (3) We propose a reduction algorithm to improve the computational efficiency of the unified framework; and (4) We present new experiment results which validate the effectiveness of the proposed methods.

The rest of the paper is organized as follows: Sec. 2 introduces the key concepts and formally defines the problem; Sec. 3 provides a brief review of hotspot detection and the NN-scan from our previous work [48] long with a new analysis of NN-scan's limitations; Sec. 4 presents the proposed unified framework, the baseline algorithm and the reduction algorithm; Sec. 5 describes the experiments and results; Sec. 6 discusses several design decisions of the proposed approach; Sec. 7 provides an extended review of literature on hotspot detection; and Sec. 8 concludes the paper and discusses the future work.

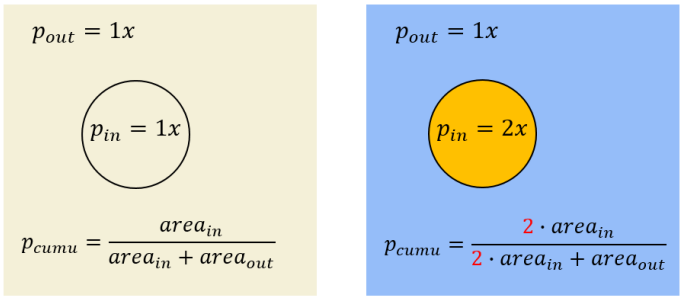
2 PROBLEM DEFINITION

2.1 Key Concepts

Point distribution: A collection of N geo-located point instances (e.g., crime cases) in a spatial domain (i.e., the study area used in a hotspot detection task).

Point process: A statistical process that generates a point distribution. It determines the probability of each point being located at each location in the study area.

Homogeneous point process (null hypothesis H_0): A process whereby the probability density of generating point instances is identical across all locations in the study area. As shown in Fig.



(a) Homogeneous point process, H_0 (b) Biased point process, H_1

Fig. 1. Homogeneous and biased point processes.

1(a), the cumulative probability of a point being generated/placed inside the circular candidate is $p_{cumu} = \frac{p_{in} \cdot area_{in}}{p_{in} \cdot area_{in} + p_{out} \cdot area_{out}} = \frac{area_{in}}{area_{in} + area_{out}}$, where $p_{in} = p_{out}$ in a homogeneous point process.

Biased point process (alternative hypothesis H_1): A process whereby the probability density of generating point instances is higher in some sub-regions (i.e., hotspots) than the others. As shown in Fig. 1(b), the cumulative probability of a point being generated/placed inside the circular candidate is $p_{cumu} = \frac{p_{in} \cdot area_{in}}{p_{in} \cdot area_{in} + p_{out} \cdot area_{out}} = \frac{2 \cdot area_{in}}{2 \cdot area_{in} + area_{out}}$, where $p_{in} = 2 \cdot p_{out}$ in a biased point process.

Hotspot: A sub-region within the study area that has a higher probability density of generating certain instances (e.g., crime cases) than its outside. The existence of hotspots means that the point process is not homogeneous and is biased towards hotspot regions.

Candidate region: A sub-region that is a hotspot candidate.

Test statistic: A statistical measure that provides a summary value for a candidate region. In this context, a test statistic value can be considered as a score for a candidate. A higher score means a higher probability of being generated by the biased point process (i.e., H_1). A test statistic (e.g., density, likelihood ratio) is needed to compare and rank candidates, which are critical operations in selecting the best candidate and testing its significance.

2.2 Formal Problem Formulation

The hotspot detection problem is formally defined as follows:

Inputs:

- An N point distribution in a spatial domain;
- A test statistic;
- A significance level α (e.g., 0.01, 0.05);

Outputs:

- Hotspots (if they exist);

Objectives:

- Solution quality (e.g., precision and recall);
- Computational efficiency;

Constraints:

- False positive rate is lower than α;
- Algorithm acceleration guarantees the same solution quality.

There are two objectives in this work. For solution quality, we aim to improve the completeness of results under the constraint that the false positive rate is controlled by α (i.e., no detection in

$(1-\alpha)\cdot 100\%$ of point distributions generated by the null hypothesis). For computational efficiency, we aim to reduce the computational cost without giving in any solution quality.

3 A REVIEW AND NEW ANALYSIS OF OUR PREVIOUS WORK ON NN-SCAN

In this section, we first briefly review our previous work [48] on the nondeterministic normalization based scan statistic (NN-scan). Then, through new analysis, we identify the limitations of NN-scan to show the need for the proposed extension.

3.1 Building Blocks of Hotspot Detection

3.1.1 Overview. At a high level, hotspot detection has three main steps:

Step-1: Enumerate through candidate regions in an input point distribution and identify the best candidate;

Step-2: Test the significance of the best candidate.

Step-3: Terminate if the best candidate is not significant; Otherwise, add it to the output, remove both its spatial coverage and contained points from the data, and go back to Step-1.

The first two steps detect a single hotspot (if it exists) from the input data, and the third step enables detection of multiple hotspots. In the following, we discuss the three main building blocks of the overall process in Sec. 3.1.2 to 3.1.4.

3.1.2 Candidate Enumeration. Since a hotspot may appear at any location in a study area, we need to define an enumeration space (e.g., certain types of circles) and go through each candidate region in that space. The spatial scan statistic uses a two-point circle based enumeration space, which covers all circular regions that have one data point at the center and another on the circumference. For an N-point distribution, this corresponds to a $O(N^2)$ search space. In general, enumeration spaces are different for detection methods focusing on different shapes (e.g., ring, linear). As discussed in Sec. 1.1, we use the circle-based enumeration space given its popularity in research and applications.

3.1.3 Candidate Evaluation. Candidate evaluation is a step coupled with the enumeration process that gives a score for each candidate region. As introduced in the basic concepts, a test statistic (e.g., density, likelihood ratio) is used to compute this score, and a higher score should correspond to a higher probability of the candidate being generated by the alternative hypothesis H_1.

The score is required to compare and rank candidates in order to select the best candidate region and test its significance. Specifically, to test the significance of a detection, we need to check if a same-quality or better candidate can be generated by the null hypothesis. Thus, the score is necessary and critical in making the comparison.

3.1.4 Significance Testing. Since there is often no existing statistical model that can be used to compute the p-value of a candidate in closed-form [18, 48], Monte Carlo simulation is used to estimate the distribution of test statistic values and compute the p-value.

The Monte Carlo simulation has M (e.g., 1000, 10000) trials. In each trial, we generate a random point distribution using the homogeneous point process (i.e., null hypothesis H_0) and run the same enumeration and evaluation algorithm (i.e., Step-1 and Step-2 in Sec. 3.1.1) to find the best candidate. Then, we insert the score of the best candidate into a table in descending order. Upon completion of all the trials, we have a table of M best scores. The detected hotspot from the real input data is significant if its score is among the top $\alpha \cdot M$ scores, where α is the significance level.

3.2 A Test Statistic with Awareness of Spatial Nondeterminism

Candidate regions may have different areas (e.g., determined by the radii of circular regions) as well as different numbers of points, so a direct ranking among them is difficult without a unifying score.

From this perspective, a test statistic can be considered as a normalizing function that generates such scalar scores and makes candidates with different areas comparable.

An ideal test statistic should guarantee the fairness among candidates of different areas (e.g., circular regions with different radii). Because a true hotspot can be of any area, any bias towards smaller or larger candidates will affect the method's reliability.

3.2.1 Existing Test Statistics and Their Problems. Since we are interested in finding regions with significantly high point concentrations, the first test statistic that comes to mind is often density $d = n/a$, where n and a are the number of points and the area of a candidate region, respectively. However, studies [31, 48] have shown that density has a strong bias towards candidate regions with smaller areas, limiting its ability to find hotspots of general areas.

To mitigate the strong bias, the spatial scan statistic (e.g., [8, 9, 16, 18, 21, 23, 32, 40]) provides a likelihood ratio based test statistic:

$$LR = \frac{Likelihood(H_1)}{Likelihood(H_0)} = \left(\frac{n}{e}\right)^n \left(\frac{N-n}{N-e}\right)^{N-n} \cdot I\left(\frac{n}{a} > \frac{N-n}{A-a}\right) \quad (1)$$

where N and n are the number of points in the entire study area and the candidate region, respectively; $e = N \cdot (a/A)$ is the expected number of points in the candidate region under the null hypothesis (A and a are the area of the study area and the candidate region, respectively); H_1 and H_0 are the alternative and null hypotheses; and $I()$ is an indicator function enforcing that the density inside a candidate region is higher than outside it (i.e., dense rather than sparse).

In Eq. (1), the likelihood of H_1 represents the probability of generating a candidate region with n points and an area of a using the biased point process (i.e., H_1). Similarly, the likelihood of H_0 represents the probability of generating the same candidate region under the homogeneous point process (i.e., H_0).

While the likelihood ratio aims to make fair comparisons among candidates with different areas, our previous study [48] showed that it still has a bias towards smaller candidates, making chance patterns (i.e., non-significant patterns) able to have better scores than significant patterns. In [48], we provided a theoretical proof that a chance pattern could have an extremely large likelihood ratio. The proof was based on a single-point based special case.

For this extension, we conducted new experiments to show that it is common for chance patterns to have better likelihood ratios than true hotspots. To generate chance patterns and true hotspots, we created 400 synthetic datasets, 100 using the homogeneous point process (i.e., H_0) and 300 using the biased point processes (i.e., H_1) with three different effect sizes. [2] The biased point process contained a sub-region with a radius $r = 1$ (in a 10x10 study area), whose inside probability density values were 100%, 150% and 200% higher than the density outside (Fig. 2 (b), (c) and (d), respectively). When the datasets follow H_0 (Fig. 2 (a)) or H_1 with not very large effective sizes (Fig. 2 (b) and (c)), we can clearly see that the candidates with a smaller area have a much higher chance of getting high likelihood ratios, which confirms the bias. Especially, there are clear separations between $r = 0.05$, $r = 0.5$, and $r = 1$ in Fig. 2 (a). As the inside probability density increases (Fig. 2 (b) and (c)), the likelihood ratios for $r = 1$ (as well as for $r = 0.5$) start getting closer to the likelihood ratios for $r = 0.05$ and become greater than them in a few cases. However, they are still mostly dominated by the sub-regions with smaller sizes (e.g., $r = 0.05$) due to the bias. In fact, by comparing the red line in Fig. 2 (b) or (c) with the blue line in Fig. 2 (a) (i.e., scores of the chance patterns), we can see that the true hotspots mostly have lower likelihood ratios than the smaller chance patterns in this example! This means that many true patterns can be mistakenly removed in significance testing because their scores may not be better than the chance patterns'. Finally, in Fig. 2 (d), where

[2]Effect size determines how many times the inside probability density is as high as outside.

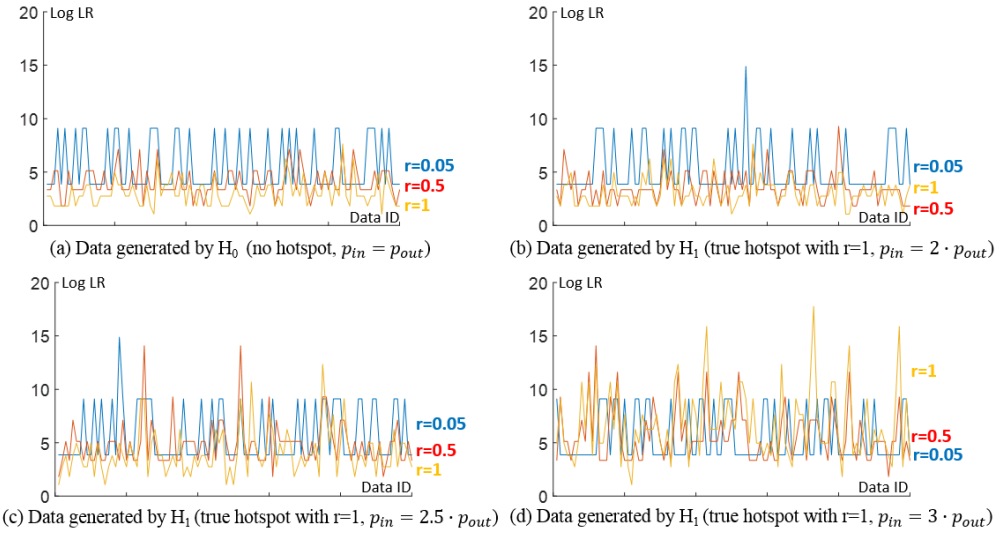


Fig. 2. Log likelihood ratios of best candidates with different areas across 100 datasets following H_0 and 300 following H_1 with different p_{in}/p_{out} ratios.

the probability density values inside the true hotspots are far greater (i.e., two times more) than the outside, the true patterns start to offset and overcome the bias, making the true hotspots more likely to be detected.

3.2.2 A Nondeterministic Normalization Index [48]. The main issue with the likelihood ratio is that it ignores the effect of spatial nondeterminism:

Definition 3.1. Spatial nondeterminism is the phenomenon that the location of the best candidate with an area a is nondeterministic in a random point distribution.

Def. 3.1 shows that it is not correct to presume that the best candidate of area a will appear at a specific location *loc*. However, in the current likelihood ratio, $Likelihood(H_0)$ is the probability of observing n points in a circular region of area a centered at a fixed location (i.e., the same as the location of the candidate being currently enumerated in the real data) [48]. As a result, it ignores the probability that a same-quality or better candidate may be generated at another location.

Our previous work [48] addresses this problem by proposing a Nondeterministic Normalization Index (NNI). For simplicity of illustration, Eq. (2) shows a simple instance of NNI.

$$NNI = \frac{n}{n^*} \tag{2}$$

$$n^* = \max_x x, \textbf{ s.t. } p(x, a) \geq \alpha \tag{3}$$

where n and a are the number of points and area of a candidate region, α is the significance level, $p(x, a)$ is the probability of observing at least one candidate region of area a with x or more points (may appear at any location) in a point distribution under H_0, and n^* is the best we can get under the null hypothesis and significance level α.

The **key parameter** and difference maker in the NNI is n^*, which gives the best we can get with the null hypothesis under a desired significance level α. The definition of n^* (Eq. (3)) explicitly considers the effect of spatial nondeterminism.

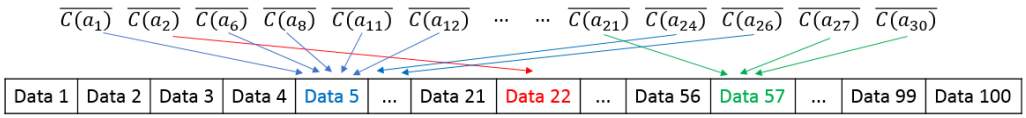


Fig. 3. Example of the non-overlapping issue in NN-scan.

3.2.3 Significance Testing with NNI. Since the significance level α is already incorporated into the definition of NNI as shown in Eq. (2) and (3), we use its value directly to determine the significance of a candidate [48]. Specifically, a candidate is considered significant if its NNI is greater than 1; otherwise, it is not significant.

3.3 Limitations of NN-scan

The main goal of NN-scan is to improve the completeness of results by avoiding the issue of chance patterns having higher scores than true hotspots. While it is able to greatly improve the success rate of hotspot detection when hotspots do exist (i.e., data belonging to H_1), our new investigation finds that its robustness against false positives is not as stable when data belongs to H_0.

In NN-scan, the computation of the Nondeterministic Normalization Index (NNI) requires Monte Carlo simulation because n^* cannot be computed in closed-form, especially considering the effect of spatial nondeterminism. In addition, n^* is area-specific so it needs to be computed for each different area a. As a result, if we still use the enumeration space of two-point circles (Sec. 3.1.2), the Monte Carlo simulation needs to be performed $O(N^2)$ times in the worst case scenario, which can be extremely expensive. Thus, in NN-scan, we reduce the enumeration space by only considering circular regions of k different radii, which brings the number of Monte Carlo simulations down from $O(N^2)$ to k.

3.3.1 Solution Quality. In our new investigations, we find that the robustness of NN-scan against false positives can be affected by k. Especially, the robustness becomes unstable for large k values, making it possible for NN-scan to output false positives in more than $100\alpha\%$ of data following H_0.

The significance level α in NNI (Eq. (2) and (3)) guarantees that less than $100\alpha\%$ of data following H_0 can contain a candidate of area a that has a NNI greater than 1 (we denote such a candidate as $\overline{C(a)}$). However, the $100\alpha\%$ of data that contain $\overline{C(a_1)}$, $\overline{C(a_2)}$, ..., or $\overline{C(a_k)}$ may not fully overlap with each other, although in practice there is often **a strong correlation** between those different $100\alpha\%$ of data for different areas. [3] As a result, this may lead to detections of false positives in more than $100\alpha\%$ of data, violating the significance level constraint.

Fig. 3 shows an example of the above analysis, where the significance level $\alpha = 0.01$. In the 100 datasets generated by H_0, $\{\overline{C(a_i)}\}$ of 20 different areas spreads over three datasets, resulting in a false positive rate of 0.03 which is above the significance level. Note that some areas may not have a $\overline{C(a)}$ in the 100 datasets since the expected number of $\overline{C(a)}$ for each area is smaller than $\alpha \cdot 100 = 0.01 \cdot 100 = 1$ in this specific example. In addition, a larger k is more likely to incur a false positive rate that is higher than the significance level. In this extension, we aim to avoid significance testing across multiple areas to enforce the constraint on the significance level.

3.3.2 Computation. The other limitation is that the algorithm acceleration in our previous study [48] was specifically designed for significance testing across multiple areas. Its idea is to use the already-computed n^* values of a subset of areas to create tight upper and lower bounds on the

[3] Increasing the number of points in a candidate at location loc also increases the number of points in the other candidates (i.e., candidates with bigger and smaller areas) at or near loc

Table 2. Pros and cons of the two approaches

Method	Pros	Cons
Likelihood ratio based	Does not require significance testing for multiple areas in a predefined set	Many true hotspots have lower scores than chance patterns due to bias, causing many missing detections (Fig. 2)
Nondeterministic normalization based	Avoids the bias across areas and improves the completeness of results	Performs significance testing for each of the multiple areas in a predefined set, causing unstable control of false positive rate (Fig. 3)

n^* values for the remaining areas, so that we may not need to compute exact n^* values for them. However, since our improved approach aims to avoid the need for multiple areas, this acceleration strategy is no longer a fit and new computational enhancements are needed.

4 A UNIFIED FRAMEWORK

We propose a unified framework that takes advantage of the strengths of both the likelihood ratio based and the Nondeterministic Normalization Index (NNI) based methods while mitigating the limitations of each. In the following, we first describe the new framework with a baseline two-phase algorithm. Then, we propose a reduction algorithm to improve the computational efficiency of the unified framework, and analyze the time complexity of the algorithms.

4.1 A Baseline Two-Phase Algorithm

Table 2 summarizes the pros and cons of the likelihood ratio based and the NNI based methods. The main issue with the likelihood ratio based approach is its unfairness across areas during significance testing (e.g., Fig. 2), which makes many true hotspots have lower scores than chance patterns and leads to missing detections. In contrast, the NNI based approach avoids the bias of likelihood ratio by separating the tests for multiple areas. However, this leads to instability in its robustness against false positives, especially when the number of areas considered is large. Based on this analysis, we aim to develop an approach which:

- Avoids the bias across areas and improves the completeness of detection compared to the likelihood ratio based approach;
- Enforces the constraint on the input significance level and avoids NNI's unstable control of the false positive rate.

To meet the two criteria, we propose a unified framework which integrates the pros of the likelihood ratio based and NNI based methods into two phases of a single approach. Fig. 4 shows the key steps of the two methods as well as the unified framework. As we can see, the likelihood ratio based method mixes up the likelihood ratios of candidates with different areas in both the observed (real) dataset and each of the simulated datasets in Monte Carlo simulation. The significance testing is then performed by ranking the best likelihood ratios from all the datasets. The NNI based method, on the other hand, separates the ranking of scores for each different area in both the observed data and each simulated data. The p-value of each candidate is then computing separately for each area.

Finally, the unified framework uses the likelihood ratio only to select the best candidate in the observed data. Its significance testing is performed using NNI, which focuses on a single area (i.e., the area of the best candidate) and does not mixes up candidates of different areas to avoid potential

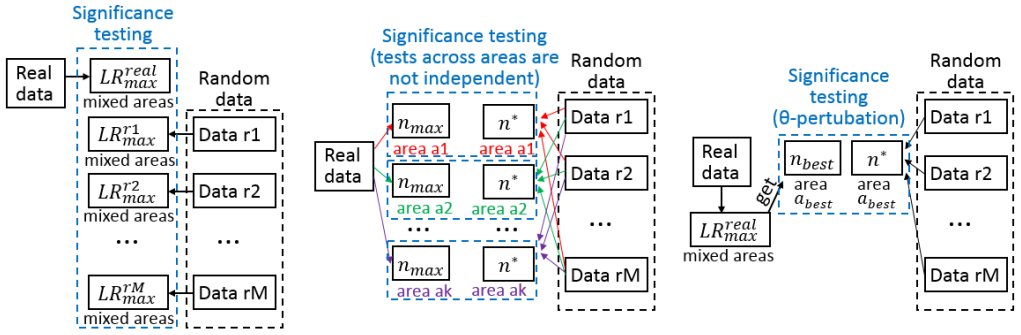


Fig. 4. Comparison of the general architectures of the three methods (best with color).

biases. These two phases of the unified framework are detailed in Sec. 4.1.1 and 4.1.2. In addition, as shown in Fig. 4, we use a θ-perturbation to further improve the method's robustness against false positives. This θ-perturbation will be discussed in Sec. 4.1.3.

4.1.1 Phase-1: Candidate Enumeration with Likelihood Ratio. In the first phase, we select the best candidate in the observed (real) point distribution using the likelihood ratio. Specifically, we enumerate through candidate regions of different areas and compute their likelihood ratio values. The candidate with the highest likelihood ratio score will be selected as the best candidate.

Since the likelihood ratio based method does not require a pre-defined set of k areas, the enumeration space of this phase can be flexible and not limited to candidates of k pre-defined areas.

In the baseline algorithm, we use the same default enumeration space as the spatial scan statistic, which covers all circular regions that have one data point at the center and another on the circumference. In other words, for each of the N data points, the algorithm enumerates $N-1$ circular regions using the data point as the center and its distance to another data point as the radius. This gives an enumeration space of $O(N^2)$.

For candidate evaluation, we need to further compute the number of points in a candidate region in order to calculate its likelihood ratio (Eq. (1)). This can be done with a brute-force linear scan which requires another $O(N)$ time. In the baseline algorithm, we make this faster by sorting. Specifically, for each center point, we sort all the points in ascending order based on their distances to the center point. After the sorting, the number of points in each candidate region can be computed in $O(1)$ time. For example, denote P_c as the current center point, and P_r as another point that has a rank of R in the sorted list (ascending order). Then, the number of points in the candidate region $CR = \{\text{center: } P_c\text{; radius: } ||P_c - P_r||_2\}$ is R. This brings down the total time complexity from $O(N^3)$ to $O(N^2 \log N)$ for Phase-1 of the baseline algorithm.

4.1.2 Phase-2: Significance Testing with Nondeterministic Normalization Index. The output of Phase-1 is the best candidate in the observed data. In the second phase, we test the statistical significance of this best candidate using the Nondeterministic Normalization Index (NNI).

Denote n and a as the number of points and area of the best candidate region from Phase-1. We calculate the n^* value in NNI (e.g., the NNI instance shown in Eq. (2)) using the Monte Carlo simulation. Again, the best candidate is significant if its $NNI > 1$; otherwise, not significant. Since Phase-2 only needs to consider a single area a, this significance testing basically checks if the probability of generating a same-quality or better candidate (i.e., a candidate of area a with at least n points) in a random data following H_0 is smaller than the significance level α.

The time complexity of each Monte Carlo trial in Phase-2 is $O(N^2)$, which is smaller than the complexity $O(N^2 \log N)$ of Phase-1 because only one area a (i.e., the area of the best candidate from Phase-1) is considered. With a total of M trials, the complexity of Phase-2 is $O(MN^2)$.

Trial parallelization: Given that the trials in Monte Carlo simulation are independent from each other, we parallelize them across λ CPU cores. This brings the complexity down to $O(\frac{MN^2}{\lambda})$.

4.1.3 A θ-Perturbation for Robustness against False Positives. Since Phase-2 only performs significance testing for a single area rather than multiple areas, it no longer has the non-overlapping problem described in Fig. 3, Sec. 3.3. While this design seems able to avoid the possibility of an increased false positive rate, our study finds that it is still not fully sufficient to enforce the significance level constraint.

To illustrate the issue, we first discuss a fully unbiased scenario. Suppose we are only interested in a single area a_{pre}, which is pre-defined before observing the input point distribution. Then, if we find the best candidate of area a_{pre} in the input data and test its significance using the Phase-2 approach, we can guarantee that the false positive rate is fully controlled by the significance level α. This scenario is unbiased because a_{pre} is not informed by any knowledge about the input data and is determined independently. However, this is not the case for the unified framework, in which the area a used in significance testing is directly determined by the best candidate in the input data. Unlike the fully unbiased scenario, a test based on this informed area a increases the chance of the detection getting a low p-value and satisfying the significance level.

Since the level of bias may depend on both the choice of test statistic and the enumeration algorithm, there is no existing model that can be used to exactly quantify and correct it. In the unified framework, we reduce the bias by imposing perturbation to the best candidate detected in the input data. Perturbation has also been found useful in other domains for bias reduction and robustness enhancement. In Earth science and physics, for example, perturbation helps improve the reliability of forecasts by reducing the bias [43, 44]. Perturbation has also been used in machine learning methods to modify the training data and improve stability in prediction. In our approach, we propose a heuristic θ-perturbation, as described by Def. 4.1, to reduce the bias.

Denote $P_{all} = \{P_1, ..., P_N\}$ as the set of input data points. Denote Z as the best candidate found in the data, c_Z as its center point, n_Z as its number of points, and r_Z as its radius. The θ-perturbed version of Z is defined as:

Definition 4.1. A θ-perturbed candidate Z^θ maintains its center at c_Z, and has a perturbed radius of $r_Z^\theta = (1-\theta) \cdot r_Z$ and number of points $n_Z^\theta = |P_{inside}|$, where $\theta > 0$ and $P_{inside} = \{P_i \,|\, \forall P_i \in P_{all}$, s.t. $||P_i - c_Z||_2 \leq r_Z^\theta\}$.

The main idea of θ-perturbation is to perform the significance testing using a perturbed version of the best candidate rather than directly using the best candidate itself. According to Def. 4.1, a perturbed candidate Z^θ is a candidate that locates at the same center as the original candidate but with a different (smaller) radius and number of points. Then, Z^θ will be used to calculate the statistical significance. The intuition is that a small perturbation is likely to have less effect on a true pattern because the sub-region of a true hotspot still has a higher probability density than its outside. In contrast, we expect to see larger effects on spurious patterns generated by pure random chance in H_0, which are more difficult to sustain without the support of higher probability density. Our experiment results (Sec. 5) show that θ-perturbation is effective and performs consistently in different set-ups. [4] Other potential formulations of the perturbation are discussed in Sec. 6.

[4]We recommend small values for θ (e.g., 0.1, 0.05) based on the experiment results.

Algorithm 1 shows the key steps of the unified framework with pseudocode. Note that the algorithm shows the process of detecting the best hotspot, which can be used recursively to allow detection of multiple hotspot as illustrated in "Step-3" at the beginning of Sec. 3.1.

4.2 Acceleration: A Reduction Algorithm

In this section, we propose a reduction algorithm to tackle the computational bottleneck on the Monte Carlo simulation in Phase-2.

The inputs of Phase-2 in the unified framework are mainly the θ-perturbed number of points n and area a of the best candidate and the total number of points N in the study area. The idea of the reduction algorithm is to construct an upper-bound of the p-value using only a small proportion of N (e.g., 0.1%, 1%, 10%). Then, if the upper-bound of the p-value is already smaller than the significance level α, we can skip the Monte Carlo simulation using the full data size N.

Thm. 4.2 shows the upper-bound of the p-value (Phase-2) using data reduction. Since the p-value defined in Sec. 4.1.2 strictly favors candidates with more points for a given area a, the upper-bound is for p-values of dense regions (i.e., the number of points is higher than the expectation under H_0) and not sparse regions.

Theorem 4.2 (Reduction Theorem). *The p-value of a candidate Z with area a_Z and number of points n_Z in a N-point distribution, is upper-bounded by the p-value of a candidate with area a_Z (same) and number of points $\lfloor \rho n_Z \rfloor$ in a $\lceil \rho N \rceil$-point distribution, where $\rho \in (0, 1]$. The spatial domains of the point distributions are the same.*

Proof. Since there is still no known statistical models that can express the p-value calculation in closed-form [18, 48], we will focus on showing the bounding relationship between the p-values rather than explicitly writing out their exact probability functions. First, it is easy to write out a closed-form expression for the probability p_{fix} of observing a same-quality or better candidate at a fixed location in a random point distribution under H_0. We will use the expression to show that the p-value of the reduced case is higher than the p-value of the original case by showing that p^{re}_{fix} at any fixed location in the reduced case is always higher than the corresponding p^{ori}_{fix} in the original case. The p^{ori}_{fix} for the original case can be written out using the binomial distribution: $p^{ori}_{fix} = 1 - \sum_{i=0}^{n_Z - 1} \binom{N}{i} p^i (1-p)^{N-i}$, where p is the probability of a single point being randomly placed into the target candidate region at the fixed location. Similarly, p^{re}_{fix} for the reduced case is: $p^{re}_{fix} = 1 - \sum_{i=0}^{\lfloor \rho n_Z \rfloor - 1} \binom{\lceil \rho N \rceil}{i} p^i (1-p)^{\lceil \rho N \rceil - i}$. [5] According to the Central Limit Theorem, the probability for each i (e.g., $\binom{N}{i} p^i (1-p)^{N-i}$) can be tightly approximated by a normal distribution.

Thus, for both cases we have $\frac{i - Np}{\sqrt{Np(1-p)}} \sim N(0, 1)$, where $N(0, 1)$ is the standard normal distribution. Then, the only difference between the original and the reduced case is that they represent different intervals in the normal distribution. Specifically, for the original case, the bounds are 0 and $n_Z - 1$. For the reduced case, the bounds are 0 and $\lfloor \rho n_Z \rfloor - 1$. Using these bounds, the summations in the expressions of p^{ori}_{fix} and p^{re}_{fix} can be expressed as the cumulative probability inside an interval of the standard normal distribution. For the original case, the interval is $[\frac{-Np}{\sqrt{Np(1-p)}}, \frac{n_Z - 1 - Np}{\sqrt{Np(1-p)}}]$. For the reduced case, the interval is $[\frac{-\lceil \rho N \rceil p}{\sqrt{\lceil \rho N \rceil p(1-p)}}, \frac{\lfloor \rho n_Z \rfloor - 1 - \lceil \rho N \rceil p}{\sqrt{\lceil \rho N \rceil p(1-p)}}]$. As we can see, the interval of the reduced case is completely contained by the interval of the original case. Since the final probability is one minus the cumulative probability inside each interval, we have $p^{ori}_{fix} \leq p^{re}_{fix}$. □

[5] The expressions for p^{ori}_{fix} and p^{re}_{fix} basically perform two operations: (1) sum up the probability for each individual event in which the number of points i is inferior to n_Z (or $\lfloor \rho n_Z \rfloor$), and (2) remove the sum from the total probability 1.

Algorithm 1: Two-phase unified framework

Require:
- An array $P = [p_1, p_2, ..., p_N]$ of N points (input data) in a spatial domain D
- Significance level α
- Number of Monte Carlo trials M
- Value of θ in θ-perturbation: v_θ

{#Phase-1}
C_{best} = new candidate(radius: null, center: null, n: null, LR: 0) {#init the best candidate}
for p_i in P **do**
  $[P_{sort}, dis_{sort}]$ = sortPointsByDistanceTo(P, target: p_i)
  for j = 2 to N **do**
    LR = computeLikelihoodRatio(n: j, a: $\pi dis_{sort}(j)^2$, N: N, A: D.area)
    if $LR > C_{best}$.LR **then**
      C_{best}.update(radius: $dis_{sort}(j)$, center: p_i, n: j, LR: LR)
    end if
  end for
end for
{#Transition: θ-Perturbation}
C_θ = C_{best}.copy()
C_θ.update(radius: $self$.radius * (1-v_θ), center: $self$.center, n: 0, LR: null)
for p_i to P **do**
  if $||p_i - C_\theta.\text{center}||_2 \leq C_\theta$.radius **then**
    C_θ.n = C_θ.n + 1
  end if
end for
{#Phase-2}
$count_{better}$ = 0
for t = 1 to M **do**
  P_r = getRandomPointDistributionByH0(N, D)
  n_t = 0
  for p_i to P_r **do**
    n_i = 0
    for p_j to P_r **do**
      if $||p_j - p_i||_2 \leq C_\theta$.radius **then**
        $n_i = n_i + 1$
      end if
    end for
    n_t = max(n_t, n_i)
    if $n_t \geq C_\theta$.n **then**
      $count_{better} = count_{better} + 1$
      break
    end if
  end for
  if $count_{better} \geq \alpha M$ **then**
    return null
  end if
end for
return C_{best}

Here we give a very simple example to help illustrate the intuition behind the Reduction Theorem. Consider an instance for Thm. 4.2 where $N = 20$, $n_Z = 20$, $\rho = 0.1$ and $a_Z = 1$ (in a 10x10 study area). In this instance the candidate contains all the points in the distribution. The theorem basically says that probability-wise it is **easier** to observe a 2-point candidate of area $a_Z = 1$ in a 2-point distribution under H_0 than to observe a 20-point candidate of area $a_Z = 1$ in a 20-point distribution under H_0. The intuition behind this example is that, when the probability of observing an event is low, the probability of observing it at a larger magnitude is even lower.

Based on Thm. 4.2, we can compute upper-bounds of the p-value using a small proportion of the full data size N. Since the bounds are tighter for a larger proportion ρ, the reduction algorithm sequentially checks through a sorted list (ascending) of ρ values and terminates as soon as an upper bound passes the significance test. Alg. 2 shows the high-level steps of the reduction algorithm.

Algorithm 2: Reduction Algorithm

Require:
- Total number of points N and the spatial domain D
- Target candidate's number of points n and radius r
- Significance level α
- Number of Monte Carlo trials M
- List L_ρ containing ρ values to consider during reduction (sorted in an ascending order)

1: **for** id = 1 to $|L_\rho|$ **do**
2: $N_{reduce} = \lceil L_\rho(id)N \rceil$
3: $n_{reduce} = \lfloor L_\rho(id)n \rfloor$
4: $is\_significant$ = MonteCarloTest(n_{reduce}, r, N_{reduce}, D, α, M) {#returns 0 or 1 using Phase-2 scheme}
5: **if** $is\_significant$==1 **then**
6: **return** $is\_significant$
7: **end if**
8: **end for**
9: $is\_significant$ = MonteCarloTest(n, r, N, D, α, M) {#upper-bounds > α, requiring full test}
10: **return** $is\_significant$

Since the reduction algorithm focuses on input data transformation, it can be used without modification with any other acceleration techniques (e.g., indexing) that are potentially useful in Monte Carlo simulation. We skip detailed discussion of these potential integrations since they are not the focus or contribution of this work. In addition, the reduction algorithm does not require any additional data structure and is very easy to implement.

4.3 Complexity Analysis

Denote the total number of points as N, the number of Monte Carlo trials as M, and the number of CPU cores as λ. To reflect the computational improvement of the reduction algorithm, we denote the working ρ (Thm. 4.2 and Alg. 2) as ρ^*, whose value is in the range $(0, 1]$. Note that Alg. 2 terminates as soon as ρ^* is found. When $\rho^* = 1$, it means that the reduction did not work and the Monte Carlo simulation needs to be executed on the full data size N. In practice, we find that ρ^* (if a true hotspot exists) is typically between 1% (when the hotspot has a very low p-value) and 10%. The total complexity of the baseline algorithm is $O(N^2 \log N + \frac{MN^2}{\lambda})$. Since $\frac{M}{\lambda}$ is much larger than $\log N$ in the vast majority of scenarios, the complexity can be simplified to $O(\frac{MN^2}{\lambda})$. The time

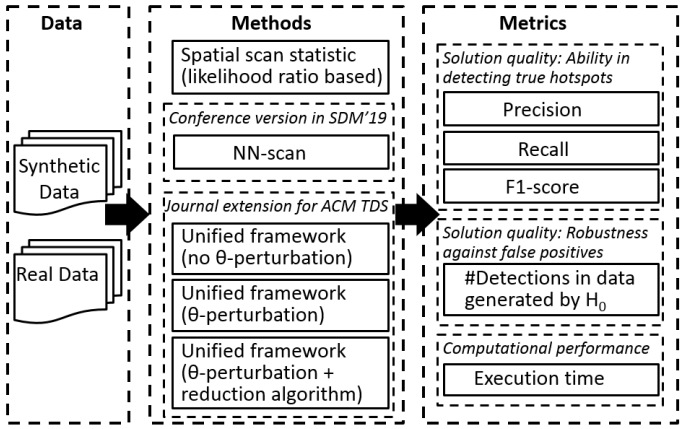


Fig. 5. The overall validation framework.

complexity of the accelerated algorithm (i.e., with the reduction theorem) is $O(N^2 \log N + \frac{M(\rho^* N)^2}{\lambda})$, where $\rho^* \in (0, 1]$. In the accelerated algorithm $\frac{M\rho^2}{\lambda}$ can be potentially smaller than $\log N$ (e.g., when $N = 1000$, $M = 1000$, $\rho^* = 10\%$, $\lambda = 10$). Note that when the upper bound does not help, the reduction algorithm may introduce additional overhead. To control the overhead to a very small amount of time, we recommend limiting the maximum ρ candidate to a small value (e.g., 10%).

5 VALIDATION

Fig. 5 shows the overall validation scheme. The goal of the experiments is to answer the following questions:

- Solution quality related:
  - Does NN-scan's robustness against false positives reduce as the number of areas considered gets larger?
  - Is the unified framework able to enforce the input significance level α (i.e., having false positive rate lower than α)?
  - Does θ-perturbation improve the unified framework's robustness against false positives?
  - Is the unified framework able to improve the completeness of results compared to the likelihood ratio based approach?
  - What are the effects of data parameters (e.g., number of points, number of hotspots, hotspot size) on the solution quality of the methods?
- Computational performance related:
  - Does the reduction algorithm improve the computational efficiency compared to the baseline algorithm?
  - What are the effects of parameters (e.g., number of points, number of Monte Carlo trials) on execution time?

5.1 Solution Quality

Since hotspots have a concrete mathematical definition (i.e., probability density of generating instances is higher within hotspots), the solution quality of hotspot detection can be experimentally evaluated using controlled synthetic data. Specifically, we generated two types of point distributions: (1) point distributions following a homogeneous point process H_0, which does not contain any

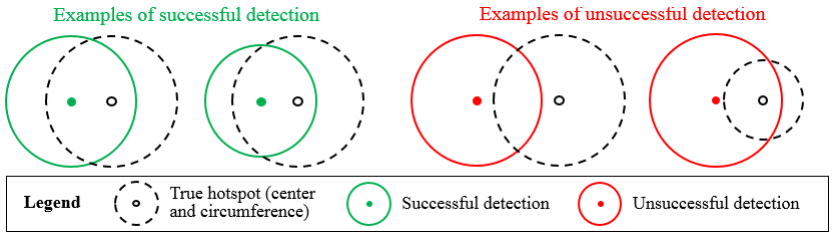


Fig. 6. Examples of successful and unsuccessful detections based on criteria defined in the experiment set-up.

hotspot; (2) point distributions following biased point processes H_1, in which we artificially insert true hotspots so that we know their actual locations and sizes. Data following H_0 evaluates if an algorithm can enforce the significance level so that it is robust against false positives, and data following H_1 evaluates how well an algorithm can detect true hotspots.

Experiment set-up: Our experiments covered a variety of parameters for the synthetic data generation: (1) Total number of points N; (2) Effect size es, which indicates how many times the probability density inside is as high as outside; (3) Radius of hotspot r; and (4) Number of hotspots h. The default parameter values of (N, es, r, h) were set to $(200, 3, 1, 2)$, respectively. The dimension of the study area was 10×10. Note that we did not specify the exact geo-distance per unit-length for the study area (i.e., the real-world distance represented by $1/10$ of the side-length of the study area). Users are free to select their own metric (e.g., meter, kilometer) or scale if visualizations on maps are needed. In order to compute the performance statistics (e.g., number of false positives, precision, recall), we need a measure to determine if a detection is successful. In the case of point distributions generated by the null hypothesis H_0, this is relatively easier because all detections will be false as there is no true hotspot. For point distributions generated by H_1 where true hotspots exist, we determine the success of detection by comparing the two key circle parameters (i.e., center location and radius) between the detection and true hotspot. Denote H_d as the detection (i.e., a circle) with center c_d and radius r_d, and H_t as the true hotspot with center c_t and radius r_t. A detection is considered successful if both the following criteria are satisfied: (1) the detection contains the center of the true hotspot, i.e., H_d.Contains(c_t)=TRUE; [6] and (2) the difference between the radii of the detection and true hotspot is smaller than half of radius of the true hotspot, i.e., $|r_d - r_t|/r_t < 1/2$. In general, it is very challenging for a method to precisely capture the center and radius of a hotspot due to the randomness involved in the data generation process. Fig. 6 visualizes several examples of successful and unsuccessful detections based on the criteria. We used the same criteria to determine the success of detection for all methods in the experiment, and then used the results to compute the performance statistics (e.g., precision, recall). In the final results, each performance statistic was calculated using detection results from 100 synthetic datasets.

In the following solution quality analyses, the results of the unified framework were obtained using the accelerated algorithm (i.e., with the reduction technique described in Sec. 4.2). During the experiments we also obtained the results of the baseline algorithm on the same datasets and the results are almost exactly the same (tiny differences are tolerable because the p-values are calculated using the Monte Carlo method which involves a little randomness). In the plots we only show the results of the accelerated version to avoid redundancy (e.g., many overlapping lines).

[6]The center of a spatial hotspot is also considered as a meaningful feature in many applications (e.g., potential source of water contamination or residence location of a serial criminal). Having the true center contained in a detection may help refine the scope for subsequent analyses on detected hotspots.

5.1.1 Robustness against false positives. In the context of hotspot detection, false positives are detections of chance patterns which are created by a homogeneous point process H_0. Fig. 7 and Fig. 8 show the experiment results on 800 synthetic datasets (100 for each pair of N and α) where point distributions were generated using H_0 (i.e., no hotspots). It offers a comparison on the robustness against false positives across different frameworks (i.e., NN-scan, Unified (no θ-perturbation), Unified (with θ-perturbation), and Likelihood ratio based). The Y-axes in Fig. 7 and Fig. 8 represent the number of datasets in which false positives were detected by a method. The significance level for all methods was set to $\alpha = 0.01$ in Fig. 7 and $\alpha = 0.05$ in Fig. 8. Since 100 datasets were used in the evaluation for each pair of N and α, a robust method is expected to detect false positives in only about one of the 100 datasets for $\alpha = 0.01$ (Fig. 7) and five for $\alpha = 0.05$ (Fig. 8). The significance levels are visualized using red dashed lines in the figures. Note that small variations are tolerable since the p-values of the methods are based on the Monte Carlo method.

Instability in NN-scan's robustness against false positives: As shown in Fig. 7 and Fig. 8, NN-scan detected more false positives than the expected number (i.e., 1 for Fig. 7; 5 for Fig. 8), especially when the number of areas k is large. For $k = 4$ and $\alpha = 0.01$, the average number of mistakes is three across the four experiments (i.e., $N = 100$, 200, 500 and 1000). This corresponds to a false positive rate of 0.03, exceeding the input significance level by 0.02. Under the same α, the false positive rate goes up to 0.05 as k increases to 20, exceeding the significance level by 0.04. A similar trend can be seen in Fig. 8. These results show that the robustness of NN-scan is affected by the number of areas considered, and thus it may not be able to fully enforce the significance level. In practice, this issue can lead to a greater number of costly false alarms (e.g., falsely declaring a normal neighborhood as a crime hotspot).

Stabilized robustness with the unified framework and the θ-perturbation: First, we can see that in both Fig. 7 and Fig. 8 there is a sharp decrease on the false positive rate from NN-scan to the unified framework. For the unified framework without θ-perturbation, the average false positive rates drop to 0.02 in Fig. 7 and 0.625 in Fig. 8 across all data sizes. These are fairly close to but still above the specified significance levels. After θ-perturbation is applied, we can clearly see that the number of mistakes is well-controlled and the average false positive rate becomes less than the specified significance levels in both figures. In addition, the robustness stays stable across different data sizes. Comparing the three unified frameworks with different θ values, the false positive rate also stays stable. Fig. 7 and Fig. 8 experimentally confirm that the unified framework and θ-perturbation are very effective in enforcing the significance level constraint.

5.1.2 Improving the completeness of results without instable control of false positive rates. Fig. 9 to 12 show the precision, recall and F1-scores achieved by the candidate methods on the synthetic datasets generated using H_1, in which true hotspots were artificially inserted using controlled parameters. In this context, improving the completeness of results means increasing the number of true hotspots being successfully detected (i.e., recall). While NN-scan achieved the highest recall among all the methods, our analysis in Sec. 3.3 and results in Sec. 5.1.1 showed that it tends to **violate the false positive rate** specified by the significance level α for data following H_0. Thus, its higher recall does not come with stable control of false positive rates, making the improvements less interesting. [7] In contrast, the unified framework is able to **enforce the significance level** and maintain a stable robustness against false positives. This is very important in urban and smart-city applications in which false positives often have a high cost (Sec. 1). In Fig. 9 to 11, we can see that all the methods using the unified framework and θ-perturbation have a higher recall (i.e., percent of true hotspots detected) compared to the likelihood ratio based approach. This means that they are able to improve the completeness of the output results while satisfying the constraint

[7]Note that we do not know in advance whether an input dataset follows H_0 or H_1.

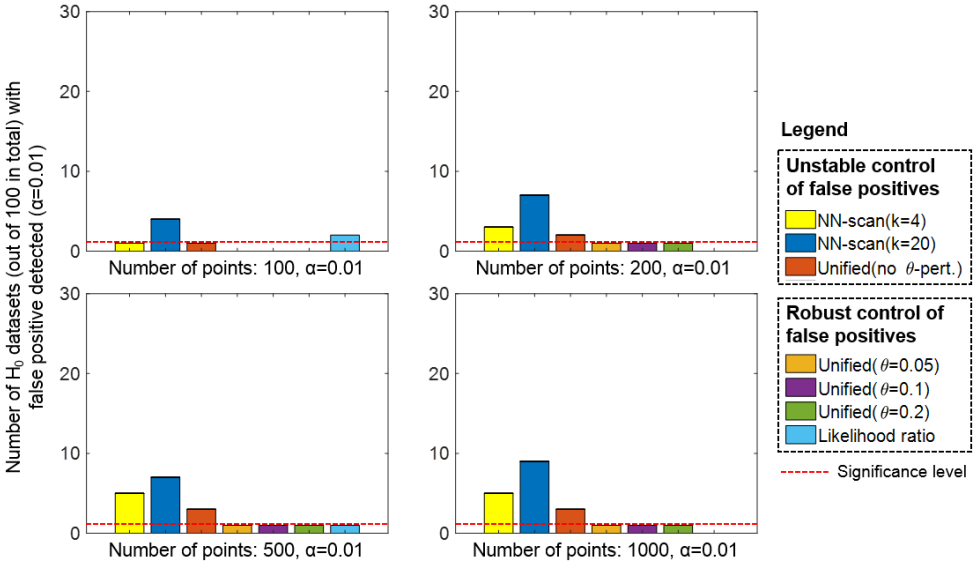


Fig. 7. Number of random datasets under H_0 (out of 100 in total) in which false positives were detected. Significance level α was set to 0.01.

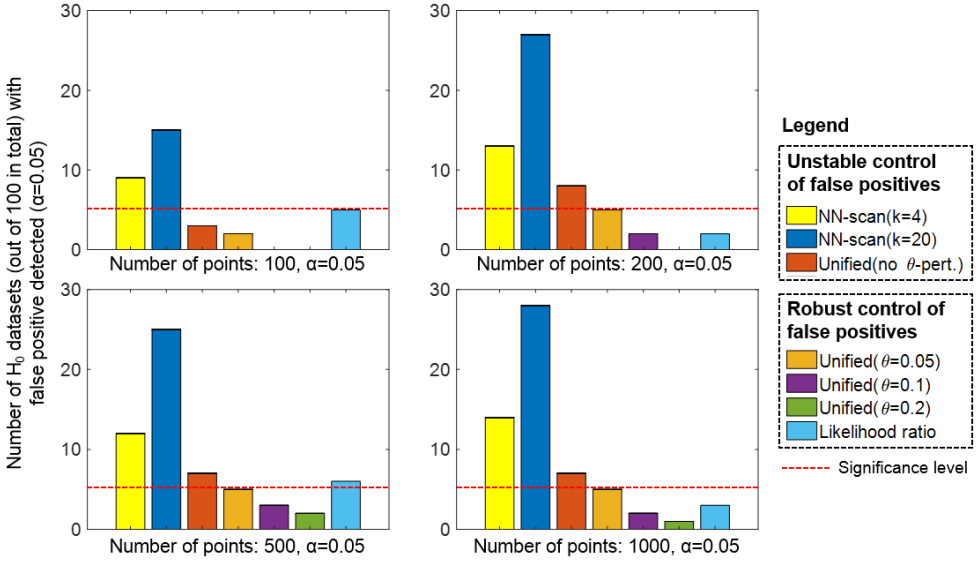


Fig. 8. Number of random datasets under H_0 (out of 100 in total) in which false positives were detected. Significance level α was set to 0.05.

on false positive rate. Among these robust methods, the best results are achieved when $\theta = 0.05$. Compared to the likelihood ratio based approach, its recall is more than 100% (relative) higher in many scenarios.

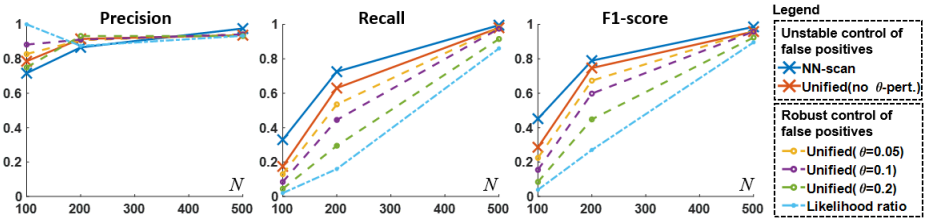


Fig. 9. Precision, recall and F1-scores with varying number of points N. Note that NN-scan and Unified (no θ-pert.) suffer from reduced robustness against false positives when **data follows H_0 (Sec. 5.1.1)**, and cannot provide guarantee on satisfying the significance level. In contrast, the likelihood ratio based approach and Unified (with θ-pert.) can enforce the significance level. Thus, the improvements in completeness achieved by Unified (with θ-pert.) are much more useful and meaningful in real-world urban or smart-city applications (e.g., [1]), where the cost of false positives is often high both socially and economically.

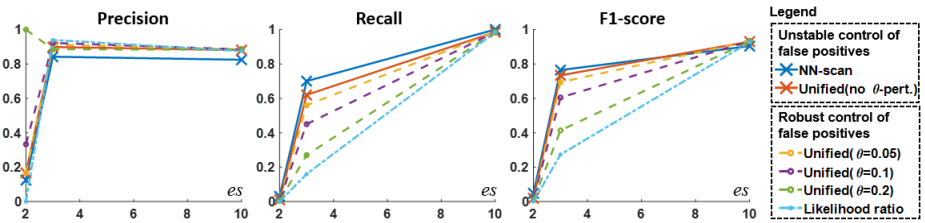


Fig. 10. Precision, recall and F1-scores with varying effect size es.

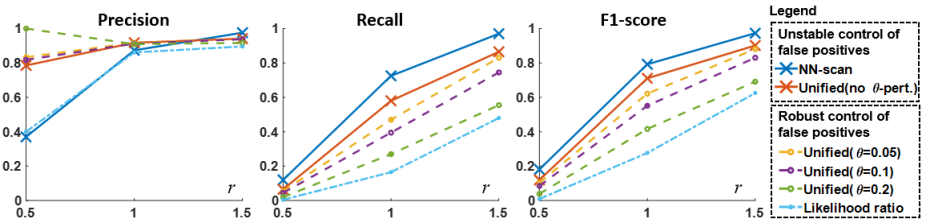


Fig. 11. Precision, recall and F1-scores with varying radius r.

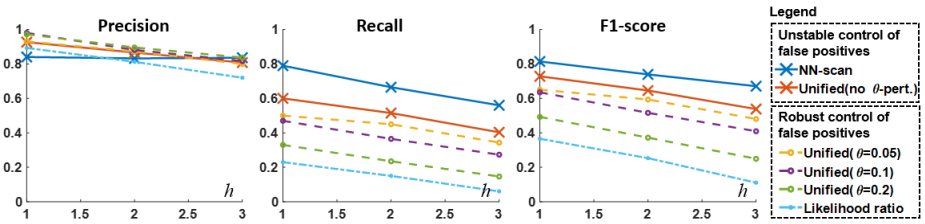


Fig. 12. Precision, recall and F1-scores with varying number of hotspots h.

5.1.3 Effect of data parameters on solution quality. Data parameters are those used to generate the synthetic data as described in the experiment set-up. Here we analyze the effect of these data parameters on the solution quality of the candidate methods.

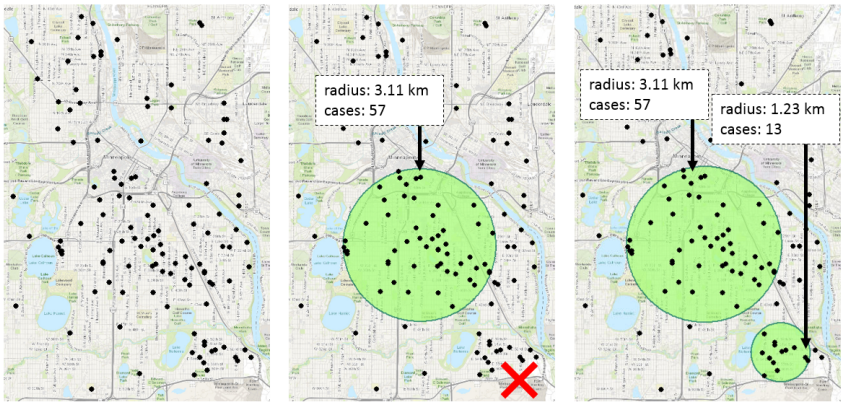


Fig. 13. Example on urban crime data: Minneapolis motor-part theft.

Effect of total number of points N**:** Fig. 9 shows that the solution quality of all the methods gradually improves as the number of points increases. This is expected as it is easier to confirm the statistical significance of hotspots with more observed samples (i.e., less random effects). For example, it can be very difficult to observe and be confident about a hotspot if we are only given one point. The confidence will grow as more point locations are given.

Effect of effect size es **and hotspot size (radius)** r**:** We discuss these two parameters together because the ways they affect the solution quality are very similar. Both effect size es and radius r determine how likely a point will randomly emerge within a hotspot under either H_1 or H_0. Since effect size es represents how many times the inside probability density is as large as the outside density, it must be greater than 1 for hotspots to exist; otherwise the point process must be homogeneous. For hotspot size r, its size determines its inside cumulative probability for a given es. If r is small, the cumulative probability may be small even with good effect sizes. In general, it can be very difficult to detect hotspots and confirm their significance when the value of es or r is small. This is consistent with the trends shown in Fig. 10 and 11, in which the overall solution quality (e.g., recall) of all methods decreases as es or r gets smaller, showing the increased difficulty for successful detection. On the bright side, however, if hotspots of small sizes are detected, we can have higher confidence in them because the effect size is likely to be very large and unusual.

Effect of number of hotspots h**:** As introduced in Sec. 3.1 (Step-3), detection methods can be used recursively as sub-routines to detect multiple hotspots. Fig. 12 shows that the solution quality of all the methods decreases as the number of hotspots increases. When multiple hotspots co-exist, it becomes more difficult to confirm the significance of a hotspot because the number of points on its outside increases due to the existence of the other hotspots.

5.1.4 Example use case: urban crime data. Since the ground-truth for this type of problems is unknown or cannot be exactly known in uncontrolled scenarios (e.g., in real-world scenarios where we do not have control over the generation process of points), the purpose here is not to offer a rigorous comparison between the solution quality of the methods. In addition, it is often difficult or insufficient to compare the methods using only one or a few datasets, which can be easily biased. Thus, in this paper, the solution quality of the methods is compared using a large number of controlled synthetic datasets (i.e., over a thousand) where the ground truth is known exactly.

The goal of this experiment is mainly to show an example of hotspot detection in a real-world urban application. The dataset used is the Motor-Part Theft dataset from the Minneapolis (USA)

Police Department. The dataset can be considered as a relatively difficult dataset for hotspot detection because the total number of points is not large, making it harder to confirm using statistical measures (Fig. 9 in Sec. 5.1.3). Fig. 13(a) shows the 124 instances displayed on top of the city map. To visually identify the dense regions, a useful trick is to compare the distances between neighboring points at different regions. With that, we can see there is a dense region around the middle of the map (i.e., covering the downtown area), and another at the bottom right (next to a lake, river park and major airport). Fig. 13(c) shows that the significance of both regions is confirmed by the unified framework. Note that the same can also be detected using the NN-scan approach [48] (i.e., the conference version) because by design it tends to return more detections. However, as discussed in Sec. 3.3 and Sec. 5.1.1, its false positive rate is not stably controlled by the significance level, making it more difficult to robustly support critical decision-making in urban applications. In contrast, the unified framework can strongly enforce the desired significance level, so we can have more confidence in its detections. As shown in Fig. 13(b), the likelihood ratio based approach confirms the larger hotspot but does not return the smaller one. This is likely due to its lower recall according to the results on synthetic datasets (Sec. 5.1.2). In addition, we expect the smaller hotspot to have a fairly large effect-size based on our analysis on the effect of hotspot size in Sec. 5.1.3. While the likelihood ratio based approach found the larger pattern but missed the smaller one, this result does not contradict the bias towards smaller candidates shown in Sec. 3.3 and Fig. 2. Since the score of a candidate depends on multiple factors such as area and effect size, the bias does not prevent larger candidates from having greater scores than smaller ones (e.g., Fig. 2 (d)). [8]

Limitation: Note that the risk of motor-part theft is not necessarily uniformly distributed in the study area, and it may depend on lots of factors such as the density of vehicles, the models of vehicles (e.g., whether anti-theft alarms are installed), availability of security cameras, risks of being noticed (e.g., by vehicles passing nearby), etc. Since these data are often not collected or publicly available, which is the case for this example, we used uniform distribution as a rough approximation of the baseline risk distribution and did not attempt to make further assumptions about it. Thus, the hotspots detected at this stage should be viewed as an aggregated result of all potential factors (including unknown ones) affecting the distribution of motor-thefts. Moreover, we mainly used the data to show what detection results may look like in real-world applications, not to suggest any policies or actions without further analysis by domain experts.

5.2 Computational Performance

We analyze the execution time of the proposed algorithms: (1) the baseline algorithm; and (2) the accelerated algorithm using reduction. The goal is to evaluate the effectiveness of the proposed reduction algorithm and identify its dominance zones. We considered these two algorithms as the candidates mainly because they are the contributions of this work and they have the same solution quality, allowing an apple-to-apple comparison. The computation and acceleration methods for the other methods are out of the scope of this work. In addition, their solution quality is different (e.g., Table 2) which will make an execution time comparison less interesting and meaningful.

5.2.1 Improvement on computational efficiency. Fig. 14 to 16 show the execution time (log scale) of the baseline and the accelerated algorithms with varying total number of points N, number of Monte Carlo trials M and effect size es. The default values of (N, M, es) were set to (1000, 500, 5). Parallelization was applied to the Monte Carlo trials for both the baseline and the accelerated

[8]The use of statistical measures in the likelihood ratio, while still having some theoretical limitations, has already greatly reduced the much stronger bias presented in other measures such as density and density ratio [48].

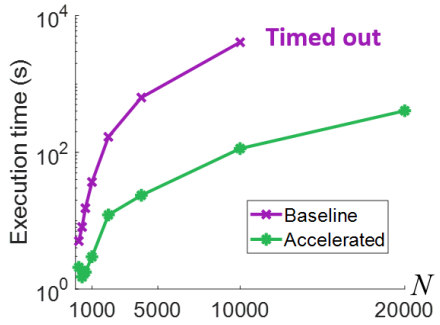


Fig. 14. Execution time with varying total number of points N.

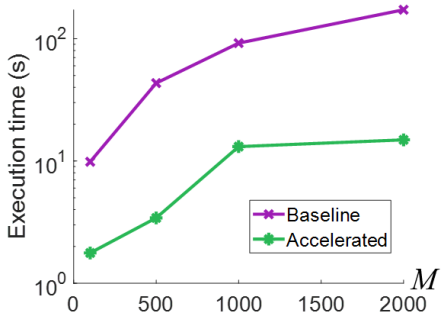


Fig. 15. Execution time with varying number of Monte Carlo trials M.

algorithms. The experiments were run using a 12-core node with an Intel Haswell E5-2680v3 processor, and each time was averaged from 10 repeated runs to reduce random effects.

Among the three controlled parameters, N and M are explicit in the time complexity expressions of the methods. We also consider effect size es because it determines how strong the true hotspots are, which may impact their corresponding p-values during detection. For a given significance level α, the p-value upper-bounds created by the reduction algorithm has a higher chance of being under α if the actual p-value is smaller.

As shown in Fig. 14 to 16, the reduction algorithm consistently outperformed the baseline algorithm in the experiments, and was orders of magnitudes faster in many of the scenarios. In general, the speed-up increases with increasing number of points N, number of Monte Carlo trials M or effect size es. In Fig. 16, we can see that the acceleration is small when $es = 2$. As discussed earlier, this is because the gap between an actual p-value and the significance level α tends to be narrower when es is small, making it more difficult for the upper-bound to satisfy the significance level. For hotspots with bigger es, the reduction algorithm is able to greatly reduce the computational cost.

Note that the reduction algorithm is currently only effective for data with true hotspots, and that its execution time is very similar to the baseline algorithm when data follows H_0 (i.e., $es = 1$). [9]

5.2.2 Effect of parameters on execution time. Here we analyze the effect of the controlled parameters, N, M and es, on the execution time.

[9]With tiny overheads when maximum ρ is set, Sec. 4.2.

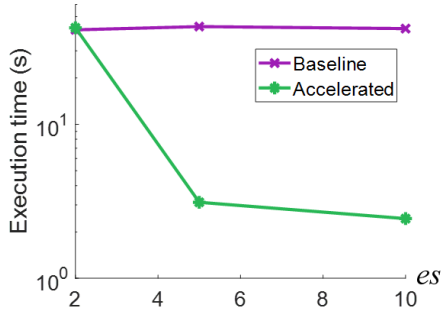


Fig. 16. Execution time with varying effect size *es*.

Effect of total number of points N**:** Fig. 14 shows the execution time trends of the baseline and the accelerated algorithms with varying N. We can see that the gap between the two algorithms increases as N increases. Specifically, the speed-up achieved by the accelerated algorithm increases from about 2 times to about 36 times as N increases from 200 to 10000. This result indicates that the working reduction factor ρ^* gets smaller as N increases. This is expected based on the solution quality results on N (i.e., Fig. 9 in Sec. 5.1.3). Basically, as N gets larger, it becomes easier to confirm the significance of true hotspots, which means the p-values of true hotspots are smaller. As a result, the reduced p-values make it easier for the upper-bounds generated by the reduction algorithm to satisfy the significance level. Similarly, smaller reduction factors also receive higher chances of success.

Effect of number of Monte Carlo trials M**:** As shown in Fig. 15, the absolute difference in the execution time between the baseline and the accelerated algorithms increases as M gets larger (i.e., from less than 10 seconds to more than 100 seconds). Overall, the accelerated algorithm has a slower increase in execution time compared to the baseline algorithm. For example, the execution time is about 7 times longer for the accelerated algorithm from $M = 100$ to $M = 2000$. For the baseline algorithm, this goes up to about 15 times. According to Sec. 4.3, the time complexity of the baseline and the accelerated algorithms are $O(N^2 \log N + \frac{MN^2}{\lambda})$ and $O(N^2 \log N + \frac{M(\rho^* N)^2}{\lambda})$, respectively, where $\rho^* \in (0, 1]$ is the working reduction factor. Although the complexity of the Monte Carlo simulation is linear in M for both algorithms, the effect of M may become secondary in the complexity of the accelerated algorithm when ρ^* is small (e.g., 0.1). This may make the total complexity "sub-linear" in M. In contrast, the complexity of the baseline algorithm is mainly dominated by the Monte Carlo simulation.

Effect of effect size *es***:** Fig. 16 shows that the computational saving achieved by the accelerated algorithm becomes greater as the effect size *es* increases. Specifically, both algorithms have very similar execution time when $es = 2$ and then the time for the accelerated algorithm quickly becomes 10-20 times faster than the baseline algorithm as *es* gets larger. As we analyzed earlier, higher effect sizes lead to lower p-values of true hotspots, which in turn leads to a higher success rate for the reduction algorithm. In Fig. 16, we can also see that the decrease in execution time becomes slower after $es = 5$. This may indicate that the reduction algorithm has already reduced the computational cost of the significance testing process (i.e., Phase-2 of the unified framework) to a level at which its time cost becomes secondary compared to the time spent on the observed data (i.e., Phase-1).

6 DISCUSSION

This section discusses three design decisions in the unified framework: significance level, θ-perturbation and hotspot shape.

Significance level: The present study considers significance level α as a hard constraint, and the goal is to improve the statistical power (i.e., recall) while satisfying this constraint. In other words, the Unified Framework aims to increase the success rate of finding true hotspots in biased point processes (i.e., H_1) under the constraint that the rate of outputting spurious patterns generated by a homogeneous point process (i.e. H_0) is smaller than α. This hard constraint is mainly favored in societal applications where false positives tend to have a high economic or social cost. As introduced earlier in Sec. 1, false claims of crime hotspots can lead to unnecessary negative impacts such as lowering property values and causing social anxiety. Limited availability of resources (e.g., budget) is also a major reason for employing this hard-constraint view. In disease monitoring and surveillance (e.g., National Cancer Institute [2]), for example, large investments are often needed in multiple areas (e.g., sanitation, screening, research, infrastructure) to study and control a disease outbreak. Thus, the inability to distinguish chance patterns from true outbreaks can lead to a huge waste of limited resources. Similarly, in transportation, new design and construction are often required to fix unsafe roads with abnormally high rates of accidents, making it critical to avoid false positives [41].

Nevertheless, we acknowledge that there are many other application areas that may favor higher success rate of detection over a strict constraint on the significance level, especially when the cost of false positives is low. In e-commerce, it may be more effective to send an advertisement to a clustered customer-group that potentially has an interest in it than to be conservative about chance patterns. In other words, the benefit of true positives (i.e., customers purchasing the recommended products/services) greatly outweighs the cost of false positives (i.e., customers skipping an advertisement that they are not interested in).

Given the non-trivial conflict between reducing false positives and increasing success rate, it may be prudent and timely to explicitly investigate this trade-off and explore ways to optimize the choice of significance level α based on budgetary constraints and quantification of benefits and costs. For example, in disease surveillance, sometimes losing control of a disease outbreak may cost more (e.g., cause more fatalities) than wasting resources on some false positives. This makes it important to develop a more holistic view in which different types of costs can be modeled simultaneously to better inform decision-making.

Choice of θ in perturbation: We have explored θ that have positive values during the perturbation. This makes the perturbed candidates have a radius smaller than the original versions according to Def. 4.1. We choose θs with positive values mainly because of the convenience they offer for a dual perturbation, i.e., simultaneously perturbing both the size of a candidate and the number of points inside it. Since an original candidate has a point on the circumference, reducing the radius by a positive proportion will also reduce the number of points by at least one. When θ is negative, it is uncertain whether the perturbation will lead to a change in the point count, especially if the radius is very small. In addition, since the perturbed candidate with a smaller radius is a sub-region of the original candidate, it maintains the higher inside probability density when the original pattern is a true hotspot, leading to relatively less effect on its significance result. A side-effect of the current θ-perturbation is that it may reduce the success rate of true hotspot detection compared to the unperturbed version. [10] Thus, in the future we will continue to refine the perturbation method to reduce this side-effect without losing its current robustness in enforcing the false positive rate. In the following, we list several opportunities for future investigations on other perturbation formulations. First, a potential richer formulation is a triple-parameter based $(\Delta n, \Delta r, \Delta d)$-perturbation in which a perturbed candidate must differ from its original form by at

[10]The unperturbed version cannot stably enforce the false positive rate constrained by the significance level, so its higher success rate is less interesting (i.e., at the cost of violating the significance level).

least Δn in points and Δr in radius, and its center must be at least Δd away from the original center. The perturbation may also be made a random process and used on all candidates instead of just the best candidate. Moreover, an orthogonal direction worth exploring is to perform perturbations on the data instead of the hotspot candidates.

Hotspot shape: This work uses circle as the shape of hotspots. This shape is by far the most common choice in both literature and applications because it has a simple mathematical representation and a relatively small enumeration space for computation. Beyond the scope of the present study, the shape of hotspots can be, in general, determined using knowledge from domain applications. For example, according to the routine activity theory [8], serial criminals (e.g., arsonists) often commit crimes neither too far nor too close to their home to reduce the travel expense as well as the risk of being recognized by neighbors. Thus, rings are better representations for the shape of hotspots than circles in this application scenario [8].

7 RELATED WORK

The spatial scan statistic [18] is the most widely used approach for hotspot detection, which defines a likelihood-ratio function to evaluate the score of each potential candidate of hotspot and test its statistical significance, as introduced in Sec. 3. In the literature, many efforts have been made to extend the capacity of the spatial scan statistic in different application scenarios. In this section, we discuss several major directions of these extensions.

Shape family: The spatial scan statistic by default detects hotspots that have a circular shape. In recent years, the shape family of hotspots has been extended to include many others such as ellipses [20, 21], rectangles [31, 32], rings [7, 8], linear [13, 35, 39, 41] or arbitrarily shaped hotspots [49]. These new hotspot shapes have mainly been developed to meet specific domain needs (e.g., ring-shaped hotspot detection for locating a serial criminal). The corresponding new techniques provide mathematical definitions of the enumeration space for hotspots of a specific shape (e.g., four-point based centric rings [8]) as well as computational models for speeding up enumeration in the new mathematical space. However, since the main focus of these methods is on the shape of the hotspot, they are still mostly built based on the likelihood ratio based framework provided by the spatial scan statistic.

Emerging hotspots: Temporal extensions have been developed to identify emerging hotspots in a spatial domain, such as the expectation-based and Bayesian scan statistics [14, 27–30, 33, 42]. While the goal in spatial scan statistics is to find stationary sub-regions that have significantly high concentration of events in data accumulated over time, emerging hotspot detection focuses on the local dynamics in the data and aims to detect sub-regions that are likely to turn into hotspots in the near-future (i.e., the pattern does not necessarily survive in long-term). One example is the expectation-based scan statistic [27], which models data as a localized time series (e.g., daily patient visits in hospitals) and computes a different form of the likelihood ratio by comparing the visits of the present day to the visits of recent days. The Bayesian scan statistic [29] also aims to detect emerging hotspots. Its main advantage is that it avoids the need for randomization tests (i.e., Monte Carlo simulation). However, it does require a correct prior or a near-approximation to make reasonable detections. While emerging hotspot detection methods have introduced new variants of test statistics (e.g., different forms of likelihood ratios), these modifications have a different purpose (i.e., identify emerging hotspots in a local time domain) and are orthogonal to the likelihood ratio formulation in the spatial scan statistic.

Multivariate hotspots: There have also been multivariate extensions of hotspot detection to allow simultaneous consideration of multiple mutually-related events. In disease surveillance, for example, a single disease can have many different symptoms (e.g., fever, diarrhea), so it may make sense to integrate a variety of symptom data at the same time to make a more powerful analysis. Multivariate

scan statistics [22, 28, 29] combine the observations of multiple types of events together in the calculation of the likelihood ratio to achieve this goal. The focus of these methods is on the design of integrating multiple likelihoods. They are not intended to modify the fundamental formulation of the likelihood ratios in univariate spatial scan statistics.

Hotspots of continuous values: The original spatial scan statistic was mainly designed for boolean events: a data point (e.g., a crime case) either exists or it does not. In contrast, continuous value based hotspot detection focuses on the values (e.g., life-span, age, air quality, house price) of points in different sub-regions rather than the density. Thus, point processes (e.g., homogeneous or biased) are no longer useful to model the statistical process. Instead, normal spatial scan statistics [10, 11, 15, 19] were developed to calculate the likelihood ratios and perform the significance tests using normal distributions. Since these methods no longer care about the density of points across the study area, the new likelihood ratios are not applicable to the boolean-event based spatial scan statistics (e.g., this work). In addition, region-size based test statistics proposed to find regions with significant changes in values [50] are also not applicable to the target problem.

Uncertainty: The effect of location uncertainty has also been studied in the context of hotspot detection [5, 6, 12, 24, 25, 34], especially for disease surveillance, where data are often shared in an aggregated form for privacy protection purposes, making it important to understand the potential errors caused by such uncertainty. Methods have been proposed to visualize the effects of data uncertainty or inaccuracy (e.g., [6, 34]).

8 CONCLUSIONS AND FUTURE WORK

We first analyzed the theoretical limitations of both the likelihood ratio based method and our previously proposed nondeterministic normalization based method (i.e., the conference version), showing that the former suffers from missing detections whereas the latter is unstable against false positives (i.e., may violate the significance level constraint). To address the limitations, we proposed a unified framework and θ-perturbation scheme, which can improve the completeness of detections over the likelihood ratio based method while still robustly enforcing the significance level constraint. We also proposed a reduction algorithm to improve the computational efficiency of the new approach. Through detailed experiments, we confirmed that the proposed approach can consistently and stably satisfy the significance level constraint, while still being able to greatly reduce the number of missing detections in the likelihood ratio based approach. In addition, execution time results showed that the reduction algorithm can greatly reduce the computational cost.

In future work, we aim to further extend the unified framework by exploring the opportunities discussed in Sec. 6 such as a combined view of precision and recall, new perturbation formulations and more hotspot shapes. In addition, while the current unified framework is designed using the continuous version of hotspot detection, we plan to extend it to handle the discrete version. Note that Phase-1 of the unified framework can mostly stay the same because likelihood ratio by default handles discrete point processes. The main change that will be needed is in the enumeration algorithm in the second phase. We also plan to extend the proposed framework to network space to better handle transportation related datasets (e.g., pedestrian fatalities, road accidents). Spatiotemporal or other higher-dimensional extensions will also be explored.

ACKNOWLEDGMENTS

This work is supported by the US NSF under Grants No. 1901099, 1737633, 1541876, 1029711, IIS-1320580, 0940818 and IIS-1218168, the USDOD under Grants HM0210-13-1-0005, ARPA-E under Grant No. DE-AR0000795, USDA under Grant No. 2017-51181-27222, NIH under Grant No. UL1

TR002494, KL2 TR002492 and TL1 TR002493 and the OVPR U-Spatial and Minnesota Supercomputing Institute at the University of Minnesota. We also would like to thank Kim Koffolt for improving the readability of the paper.

REFERENCES

[1] 2017. Connecting the Smart-City Paradigm with a Sustainable Urban Infrastructure Systems Framework to Advance Equity in Communities. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1737633&HistoricalAwards=false.
[2] 2017. National Cancer Institute, Surveillance Research Program. https://surveillance.cancer.gov//.
[3] 2017. SaTScan. https://www.satscan.org/.
[4] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. 2018. Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 83.
[5] Jose Cadena, Arinjoy Basak, Anil Vullikanti, and Xinwei Deng. 2018. Graph scan statistics with uncertainty. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
[6] Eric Delmelle, Coline Dony, Irene Casas, Meijuan Jia, and Wenwu Tang. 2014. Visualizing the impact of space-time uncertainties on dengue fever patterns. *Intl. J. of Geographical Information Science* 28, 5 (2014), 1107–1127.
[7] Emre Eftelioglu, Yan Li, Xun Tang, Shashi Shekhar, James M Kang, and Christopher Farah. 2016. Mining network hotspots with holes: A summary of results. In *Intl. Conf. on Geographic Information Science*. Springer, 51–67.
[8] Emre Eftelioglu, Shashi Shekhar, Dev Oliver, Xun Zhou, Michael R Evans, Yiqun Xie, James M Kang, Renee Laubscher, and Christopher Farah. 2014. Ring-shaped hotspot detection: a summary of results. In *2014 IEEE International Conference on Data Mining*. IEEE, 815–820.
[9] Emre Eftelioglu, Xun Tang, and Shashi Shekhar. 2015. Geographically robust hotspot detection: a summary of results. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*. IEEE, 1447–1456.
[10] Lan Huang, Ram C Tiwari, Linda W Pickle, and Zhaohui Zou. 2010. Covariate adjusted weighted normal spatial scan statistics with applications to study geographic clustering of obesity and lung cancer mortality in the United States. *Statistics in Medicine* 29, 23 (2010), 2410–2422.
[11] Lan Huang, Ram C Tiwari, Zhaohui Zou, Martin Kulldorff, and Eric J Feuer. 2009. Weighted normal spatial scan statistic for heterogeneous population data. *J. Amer. Statist. Assoc.* 104, 487 (2009), 886–898.
[12] Yan Huang and Jason W Powell. 2012. Detecting regions of disequilibrium in taxi services under uncertainty. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 139–148.
[13] Vandana Pursnani Janeja and Vijayalakshmi Atluri. 2005. Ls 3: A linear semantic scan statistic technique for detecting anomalous windows. In *Proceedings of the 2005 ACM symposium on Applied computing*. ACM, 493–497.
[14] Xia Jiang and Gregory F Cooper. 2010. A Bayesian spatio-temporal method for disease outbreak detection. *Journal of the American Medical Informatics Association* 17, 4 (2010), 462–471.
[15] Inkyung Jung, Martin Kulldorff, and Otukei John Richard. 2010. A spatial scan statistic for multinomial data. *Statistics in medicine* 29, 18 (2010), 1910–1918.
[16] Julia Krolik, Gerald Evans, Paul Belanger, Allison Maier, Geoffrey Hall, Alan Joyce, Stephanie Guimont, Amanda Pelot, and Anna Majury. 2014. Microbial source tracking and spatial analysis of E. coli contaminated private well waters in southeastern Ontario. *Journal of water and health* 12, 2 (2014), 348–357.
[17] Julia Krolik, Allison Maier, Gerald Evans, Paul Belanger, Geoffrey Hall, and Alan Joyce. 2013. A spatial analysis of private well water Escherichia coli contamination in southern Ontario. *Geospatial Health* 8, 1 (2013), 65–75.
[18] Martin Kulldorff. 1997. A spatial scan statistic. *Comm. in Statistics-Theory and methods* 26, 6 (1997), 1481–1496.
[19] Martin Kulldorff, Lan Huang, and Kevin Konty. 2009. A scan statistic for continuous data based on the normal probability model. *International journal of health geographics* 8, 1 (2009), 58.
[20] Martin Kulldorff, Lan Huang, and Linda Pickle. 2003. An elliptic spatial scan statistic and its application to breast cancer mortality data in Northeastern United States. *Journal of Urban Health* 80 (2003), i130–i131.
[21] Martin Kulldorff, Lan Huang, Linda Pickle, and Luiz Duczmal. 2006. An elliptic spatial scan statistic. *Statistics in medicine* 25, 22 (2006), 3929–3943.
[22] Martin Kulldorff, Farzad Mostashari, Luiz Duczmal, W Katherine Yih, Ken Kleinman, and Richard Platt. 2007. Multivariate scan statistics for disease surveillance. *Statistics in medicine* 26, 8 (2007), 1824–1833.
[23] Michael Leitner and Marco Helbich. 2011. The impact of hurricanes on crime: a spatio-temporal analysis in the city of Houston, Texas. *Cartography and Geographic Information Science* 38, 2 (2011), 213–221.
[24] Lan Luo. 2013. Impact of spatial aggregation error on the spatial scan analysis: a case study of colorectal cancer. *Geospatial health* (2013), 23–35.
[25] Nicholas Malizia. 2013. Inaccuracy, uncertainty and the space-time permutation scan statistic. *PloS one* 8, 2 (2013), e52034.

[26] Tomoki Nakaya and Keiji Yano. 2010. Visualising crime clusters in a space-time cube: An exploratory data-analysis approach using space-time kernel density estimation and scan statistics. *Transactions in GIS* 14, 3 (2010), 223–239.
[27] Daniel B Neill. 2009. Expectation-based scan statistics for monitoring spatial time series data. *International Journal of Forecasting* 25, 3 (2009), 498–517.
[28] Daniel B Neill. 2011. Fast Bayesian scan statistics for multivariate event detection and visualization. *Statistics in Medicine* 30, 5 (2011), 455–469.
[29] Daniel B Neill and Gregory F Cooper. 2010. A multivariate Bayesian scan statistic for early event detection and characterization. *Machine learning* 79, 3 (2010), 261–282.
[30] Daniel B Neill, Gregory F Cooper, Kaustav Das, Xia Jiang, and Jeff Schneider. 2009. Bayesian network scan statistics for multivariate pattern detection. In *Scan Statistics*. Springer, 221–249.
[31] Daniel B Neill and Andrew W Moore. 2004. A fast multi-resolution method for detection of significant spatial disease clusters. In *Advances in Neural Information Processing Systems (NIPS)*. 651–658.
[32] Daniel B Neill and Andrew W Moore. 2004. Rapid detection of significant spatial clusters. In *Proc. ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*. 256–265.
[33] Daniel B Neill, Andrew W Moore, and Gregory F Cooper. 2006. A Bayesian spatial scan statistic. In *Advances in neural information processing systems*. 1003–1010.
[34] Fernando LP Oliveira, André LF Cançado, Luiz H Duczmal, and Anderson R Duarte. 2012. Assessing the outline uncertainty of spatial disease clusters. *Public Health – Methodology, Environmental and Systems Issues* (2012), 51.
[35] Dev Oliver, Shashi Shekhar, James M Kang, Renee Laubscher, Veronica Carlan, and Abdussalam Bannur. 2013. A k-main routes approach to spatial network activity summarization. *IEEE transactions on knowledge and data engineering* 26, 6 (2013), 1464–1478.
[36] Sushil K Prasad, Danial Aghajarian, Michael McDermott, Dhara Shah, Mohamed Mokbel, Satish Puri, Sergio J Rey, Shashi Shekhar, Yiqun Xie, Ranga Raju Vatsavai, Fusheng Wang, Yanhui Liang, Hoang Vo, and Shaowen Wang. 2017. Parallel processing over spatial-temporal datasets from geo, bio, climate and social science communities: A research roadmap. In *2017 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 232–250.
[37] Shashi Shekhar, Steven Feiner, and Walid Aref. 2015. Spatial computing. *Commun. ACM* 59, 1 (2015), 72–81.
[38] Shashi Shekhar, Zhe Jiang, Reem Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. 2015. Spatiotemporal data mining: A computational perspective. *ISPRS International Journal of Geo-Information* 4, 4 (2015), 2306–2338.
[39] Lei Shi and Vandana P Janeja. 2009. Anomalous window discovery through scan statistics for linear intersecting paths (SSLIP). In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 767–776.
[40] Joanne R Stevenson, Christopher T Emrich, Jerry T Mitchell, and Susan L Cutter. 2010. Using building permits to monitor disaster recovery: A spatio-temporal case study of coastal Mississippi following Hurricane Katrina. *Cartography and Geographic Information Science* 37, 1 (2010), 57–68.
[41] Xun Tang, Emre Eftelioglu, Dev Oliver, and Shashi Shekhar. 2017. Significant linear hotspot discovery. *IEEE Transactions on Big Data* 3, 2 (2017), 140–153.
[42] Jonathan Wakefield and Albert Kim. 2013. A Bayesian model for cluster detection. *Biostatistics* 14, 4 (2013), 752–765.
[43] Clemens Wastl, Yong Wang, Aitor Atencia, and Christoph Wittmann. 2019. Independent perturbations for physics parametrization tendencies in a convection-permitting ensemble (pSPPT). *Geoscientific Model Development* 12, 1 (2019), 261–273.
[44] Antje Weisheimer, Susanna Corti, Tim Palmer, and Frederic Vitart. 2014. Addressing model error through atmospheric stochastic physical parametrizations: impact on the coupled ECMWF seasonal forecasting system. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 372, 2018 (2014), 20130290.
[45] Claire S Witham and Clive Oppenheimer. 2004. Mortality in England during the 1783–4 Laki Craters eruption. *Bulletin of Volcanology* 67, 1 (2004), 15–26.
[46] Yiqun Xie, Emre Eftelioglu, Reem Ali, Xun Tang, Yan Li, Ruhi Doshi, and Shashi Shekhar. 2017. Transdisciplinary foundations of geospatial data science. *ISPRS International Journal of Geo-Information* 6, 12 (2017), 395.
[47] Yiqun Xie, Jayant Gupta, Yan Li, and Shashi Shekhar. 2018. Transforming Smart Cities with Spatial Computing. In *2018 IEEE International Smart Cities Conference (ISC2)*. IEEE, 1–9.
[48] Yiqun Xie and Shashi Shekhar. 2019. A Nondeterministic Normalization based Scan Statistic (NN-scan) towards Robust Hotspot Detection: A Summary of Results. In *SIAM International Conference on Data Mining (SDM'19)*. SIAM.
[49] Yiqun Xie and Shashi Shekhar. 2019. Significant DBSCAN towards Statistically Robust Clustering. In *Proceedings of the 16th International Symposium on Spatial and Temporal Databases*. 31–40.
[50] Yiqun Xie, Xun Zhou, and Shashi Shekhar. in press. Discovering Interesting Sub-Paths with Statistical Significance from Spatio-temporal Datasets. *ACM Transactions on Intelligent Systems and Technology* (in press).