Significant DBSCAN+: Statistically Robust Density-based Clustering

YIQUN XIE, University of Maryland
XIAOWEI JIA, University of Pittsburgh
SHASHI SHEKHAR, University of Minnesota
HAN BAO and XUN ZHOU, University of Iowa

Cluster detection is important and widely used in a variety of applications, including public health, public safety, transportation, and so on. Given a collection of data points, we aim to detect density-connected spatial clusters with varying geometric shapes and densities, under the constraint that the clusters are statistically significant. The problem is challenging, because many societal applications and domain science studies have low tolerance for spurious results, and clusters may have arbitrary shapes and varying densities. As a classical topic in data mining and learning, a myriad of techniques have been developed to detect clusters with both varying shapes and densities (e.g., density-based, hierarchical, spectral, or deep clustering methods). However, the vast majority of these techniques do not consider statistical rigor and are susceptible to detecting spurious clusters formed as a result of natural randomness. On the other hand, scan statistic approaches explicitly control the rate of spurious results, but they typically assume a single "hotspot" of over-density and many rely on further assumptions such as a tessellated input space. To unite the strengths of both lines of work, we propose a statistically robust formulation of a multi-scale DBSCAN, namely Significant DBSCAN+, to identify significant clusters that are density connected. As we will show, incorporation of statistical rigor is a powerful mechanism that allows the new Significant DBSCAN+ to outperform state-of-the-art clustering techniques in various scenarios. We also propose computational enhancements to speed-up the proposed approach. Experiment results show that Significant DBSCAN+ can simultaneously improve the success rate of true cluster detection (e.g., 10–20% increases in absolute F1 scores) and substantially reduce the rate of spurious results (e.g., from thousands/hundreds of spurious detections to none or just a few across 100 datasets), and the acceleration methods can improve the efficiency for both clustered and non-clustered data.

Yiqun Xie is supported in part by NSF grants 2105133 and 2126474, Google's AI for Social Good Impact Scholars program, and the Dean's Research Initiative Award at the University of Maryland; Xiaowei Jia is supported in part by USGS award G21AC10207, and Pitt Momentum Fund Award; Shashi Shekhar is supported in part by NSF grants 1901099, 1737633, IIS-1320580, IIS-0940818, IIS-1218168, and 1916518, USDOD grant HM0476-20-1-0009, USDOE (ARPA-E) award DE-AR0000795, NIH grants UL1 TR002494, KL2TR002492, and TL1 TR002493, USDA grant 2017-51181-27222, and the Minnesota Supercomputing Institute; and Han Bao and Xun Zhou are supported in part by the Safety Research using Simulation University Transportation Center (SAFER-SIM), which is funded by US-DOT's University Transportation Centers Program through award 69A3551747131.

Authors' addresses: Y. Xie, University of Maryland, 1124 Lefrak Hall, 7251 Preinkert Dr., College Park, MD 20742; email: xie@umd.edu; X. Jia, University of Pittsburgh, Sennott Square Building, Room 6135, 210 S. Bouquet Street Pittsburgh, PA 15260-9150; email: xiaowei.jia@upitt.edu; S. Shekhar, University of Minnesota, 4-192 Keller Hall, 200 Union Street SE, Minneapolis, MN 55455; email: shekhar@umn.edu, H. Bao, University of Iowa, 14 MacLean Hall, Iowa City, IA 52242; email: han-bao@uiowa.edu; X. Zhou, University of Iowa, 108 John Pappajohn Business Building, Iowa City, IA 52242; email: xun-zhou@uiowa.edu.

2157-6904/2021/11-ART62 $15.00
https://doi.org/10.1145/3474842

CCS Concepts: • **Computing methodologies → Cluster analysis**; • **Information systems → Clustering;**

Additional Key Words and Phrases: Clustering, DBSCAN, statistical robustness

ACM Reference format:
Yiqun Xie, Xiaowei Jia, Shashi Shekhar, Han Bao, and Xun Zhou. 2021. Significant DBSCAN+: Statistically Robust Density-based Clustering. *ACM Trans. Intell. Syst. Technol.* 12, 5, Article 62 (November 2021), 26 pages.
https://doi.org/10.1145/3474842

1 INTRODUCTION

Detection of significant clusters are of great value to a variety of domain applications, including public health, public safety, transportation, economics, and so on. In public health, for example, epidemiologists have used significant clusters (a.k.a, hotspots) to monitor and alert the public about disease outbreaks (e.g., legionnaires' disease, leukemia) [17, 22]. The Research Surveillance Program at the National Cancer Institute has included significant clustering (e.g., SaTScan) as an important methodology and tool [2]. In public safety, police officers use clusters of crime cases to identify neighborhoods with abnormally high crime rates or locate serial criminals [12]. In transportation, many local governments (e.g., U.S. states) have launched "Zero Death" initiatives to save lives from traffic-related accidents. Clustering helps planners find roads with significantly high concentration of car accidents or pedestrian fatality, which are indicators of potentially unsafe driving conditions (e.g., damaged side walks, pot holes). In the COVID-19 pandemic, there is also an opportunity of using clustering to group local communities with similar characteristics (e.g., portfolios of COVID-19 cases, economic status, demographics) to effectively gather and share management experience and improve the policy making process for members in each group. Significant clustering in general is also an important topic in spatial data mining [6, 35, 36].

In many of these societal use cases or applications, there is often a high cost associated with spurious patterns. For example, identifying a region as a crime cluster by mistake can greatly reduce the number of people visiting the region, lower property values and hurt local businesses. Similarly, in domain science studies (e.g., Earth science, mechanical engineering), clustering techniques can assist researchers in generating interesting and non-trivial hypotheses for further field or laboratory investigation. Spurious results in such exploratory analyses may lead to a huge waste of resources and time, defeating the purpose of assisting or accelerating science discovery. For this reason, robust control of the rate of spurious results can be invaluable for further improving the value of clustering in these broad applications.

Given a collection of points in a domain (e.g., a confined geographic space, a bounded multi-dimensional space), we aim to detect clusters of arbitrary shapes and varying densities, where the points in each cluster are density-connected by a density level θ, and the output clusters are statistically significant under a specified significance level α.

Data mining and learning communities have developed a vast literature on clustering, including partition-based methods (e.g., k-means, CLARANS), local density-based methods (e.g., DBSCAN and DENCLUE), spectral graph theory based methods (e.g., normalized-cut, spectral clustering), hierarchical methods (e.g., OPTICS, CURE, Chameleon [16], and HDBSCAN [8]), local similarity methods [25, 29], two-dimensional spatial projection methods [27, 28], unsupervised feature selection [31, 47], deep clustering [15, 30, 40], and many more [45, 46]. A majority of these methods, at a high level, formulate a cluster as a spatially contiguous point set of high-density, and high-density sets disconnected by a low-density gap are considered as individual clusters. Arbitrary shapes of clusters and variations of densities across clusters are major challenges that have attracted continued and common interest in the literature. For example, hierarchical density-based clustering

(HDBSCAN [1, 7, 8, 23]) is a recent state-of-the-art developed by the original authors of DBSCAN and OPTICS to address both of these challenges, with reduced need of parameter selection. However, most clustering techniques still have not incorporated statistical robustness, and thus are more likely to yield many spurious clusters in the output [8, 20, 42, 44]. On the other hand, scan statistic approaches are commonly studied in statistics communities [3, 14, 17, 22, 34], which also aim to identify sub-regions with over-density or intensification of events. Scan statistic methods explicitly model randomness embedded in real-world datasets and incorporate significance testing to eliminate spurious clusters. However, a common assumption in scan statistics is that there exists a single over-density region in the data, and test statistics (e.g., likelihood ratios [17]) are mostly defined based on it, limiting the use of scan statistic methods in data with many clusters of varying densities (e.g., detecting multiple clusters as a single piece [18, 21]). Also unlike most clustering techniques in data mining, scan statistic approaches typically do not consider the contiguity of density within a cluster. In addition, majority of developments in scan statistics are for predefined-shape-based clusters (e.g., circular [3, 17], rectangular [22], ring [12], and linear [37]) due to extensive computational cost, and the irregular-shape extensions often rely on tessellated input spaces (e.g., county maps) and aggregated data [9–11, 26].

To unite the strengths of these two tracks of research in data mining and statistics communities, our preliminary work proposed a Significant DBSCAN approach [42], which incorporates statistical rigor into the density-based DBSCAN clustering to robustly remove spurious detections. We also developed a dual-convergence algorithm to improve the efficiency of the approach. However, Significant DBSCAN was mainly designed for single-density scenarios (i.e., one-pair of $(\epsilon, minPts)$). When it comes to clusters with varying densities, a pre-specified density list based on uniform sampling was employed, but we found the solution quality may suffer if the rigidly chosen list does not match well with the true densities. In addition, the method under-performs when moderate overlaps exist between local-density-distributions of clusters (detailed in Section 3.3).

In this extension, we propose a Significant DBSCAN+ with three contributions to address the limitations of the preliminary work. First, we propose a **one-at-a-time (OaaT)** density selection strategy to better separate and approximate the various true densities of clusters. Second, we develop a multi-scale DBSCAN sub-routine to improve the ability of Significant DBSCAN+ in correctly capturing a cluster with a wider range of input densities (i.e., reduced sensitivity). Finally, we propose a **Virtual sequence visit (VISIT)** approach to select a stable set of clusters for significance testing to reduce scattered cluster results caused by overlaps in cluster densities. Computational enhancements are also presented to improve the efficiency of Significant DBSCAN+.

Experiment results show that the proposed Significant DBSCAN+ can greatly reduce the number of spurious clusters, and meanwhile, improve the success rate of true cluster detection compared to existing clustering techniques as well as our preliminary work. In addition, the algorithmic acceleration can greatly reduce computational cost. We also create and share a new benchmark dataset containing 5,400 individual datasets under a large variety of scenarios (e.g., with and without clusters; different effect sizes of clusters) to help future evaluation and comparison.

Scope and outline: The scope of the present study is to improve the statistical robustness of clustering techniques, and to encourage the use of statistically robust extensions and formulations of clustering in the data mining community. Significant DBSCAN+ is not directly applicable to many scan statistic problems where control data (e.g., underlying population in disease surveillance) is needed. In addition, Significant DBSCAN+ favors clusters that are density-connected or density-contiguous at a density-level θ (Section 2). In other words, low-density gaps are considered as separations between clusters (same as most clustering techniques in data mining), which differs from top-down region-enumeration based approaches. The rest of the article is

organized as follows: Section 2 describes the problem definition, Section 3 summarizes our preliminary work on Significant DBSCAN, Section 4 shows the extended version Significant DBSCAN+, Section 5 presents experiment results, and finally, Section 6 concludes the article with future directions.

2 PROBLEM DEFINITION

2.1 Basic Concepts

Definition 2.1 (Density-connected Cluster C_θ). Given a density criterion θ for a local neighborhood around a point, a cluster C_θ is density-connected if any two points in C_θ are mutually reachable through a sequence of in-cluster points that satisfy θ. This implies that any point in C_θ either directly satisfies θ or is part of the local neighborhood of a point satisfying θ. In DBSCAN, a criterion θ is determined by a (ϵ, $minPts$) pair, where ϵ is the radius of a local neighborhood and $minPts$ is the minimum number of points required.

Definition 2.2 (Base Clustering Algorithm ALG_{base}). A clustering algorithm (e.g., DBSCAN or its variation in this article) that is used as the base for the incorporation of statistical robustness.

Definition 2.3 (Bounded Domain $\mathcal{D}$). A bounded sub-space of a d-dimensional space. A bound $\mathcal{D}$ can be defined by a set of closed intervals along each dimension, hyperplane or hypersphere. $\mathcal{D}$ determines the range of values allowed for points in $\mathbb{R}^d$. Examples of $\mathcal{D}$ include a two-dimensional interval specified by $[x_{min}, x_{max}, y_{min}, y_{max}]$, a city boundary, or sub-spaces within a city-boundary where certain events can be located (e.g., a traffic accident).

Definition 2.4 (Point Process). A statistical process that governs the generation of a point distribution in $\mathcal{D}$. It determines the probability or probability density of having a point at each location $loc \in \mathcal{D}$. A homogeneous point process (e.g., complete spatial randomness) has identical probability or probability density across all locations (i.e., no true cluster). In contrast, a biased/clustered point process has higher probabilities for locations inside the clusters and lower outside.

Definition 2.5 (Hypotheses H_0 and H_1). For clustering, the null hypothesis H_0 states that a point distribution in $\mathcal{D}$ is generated by a homogeneous point process (i.e., no true cluster), whereas the alternative hypothesis H_1 states that data in $\mathcal{D}$ follows a clustered point process, and there exist sub-spaces of $\mathcal{D}$ where probabilities or probability densities are higher.

Definition 2.6 (Test Statistic $\mathcal{T}$). A random variable used to summarize a set of sample data points (e.g., a set of points in a cluster C_θ) and test the hypotheses. In this context, it can be considered as a score calculated from the data (e.g., density of a cluster). The significance of the score determines whether to reject the null hypothesis.

2.2 Formal Problem Formulation

The problem is formally defined as follows:

Inputs:

- A distribution of N points in $\mathcal{D}$;
- A set of parameters S_{para}—including those related to local density criteria θ—for a base clustering algorithm ALG_{base} (e.g., DBSCAN or its variation in this article);
- A test statistic $\mathcal{T}$;
- A significance level α.

Output:

- Statistically significant clusters of arbitrary shapes and varying densities.

Goals:

- Solution quality (e.g., measured by precision, recall and number of spurious clusters);
- Computational efficiency.

Constraints:

- Output clusters satisfy significance level α. Equivalently, spurious clusters generated by H_0 are only detected in less than αM of M datasets;
- Output clusters are density-connected under criteria $\theta \in S_{para}$ (i.e., no low-density gaps);
- Output clusters are not altered by computational enhancements.

The problem definition shows the main scope of the article, which is to enable statistically robust clustering for a base algorithm ALG_{base} (e.g., DBSCAN). Note that ALG_{base} is only a sub-routine of a statistically robust formulation, whose final output is not necessarily (and often not) a subset of clusters (that are significant) from a typical ALG_{base} execution. As we will show in Sections 3 and 4, incorporation of statistical rigor enables more flexible design for the overall clustering process, and thus may serve as a potentially powerful mechanism to improve the results of a broader set of clustering techniques. For example, using the single-density based DBSCAN as ALG_{base}, the significant version is flexible in detecting clusters with varying densities while being able to remove spurious detections (Section 3.2).

3 PRELIMINARY RESULTS AND LIMITATIONS

Our preliminary work [42] explores a Significant DBSCAN to make the clustering algorithm robust against spurious patterns. In this section, we will summarize the general formulation, computational enhancements as well as limitations.

3.1 Significant DBSCAN: The General Formulation

Here we introduce the statistical modeling (e.g., test statistic) of Significant DBSCAN and show the testing procedure for DBSCAN with a single density criterion (ϵ, $minPts$). The multi-density version will be discussed next in Section 3.2.

3.1.1 Modeling of Statistical Significance. To model the statistical significance of clustering results, it is necessary to know how the clusters are searched and selected during the detection process. Thus, a base clustering algorithm is needed before statistical significance can be modeled. Our preliminary work [42] uses DBSCAN as the base clustering algorithm mainly due to its wide adoption and large user community.[1]

In Reference [42], we explore several types of test statistics for hypothesis testing as shown in Table 1. For a given cluster from a point distribution, its test statistic value will be used to determine if it can be generated by the null hypothesis under significance level α. According to Table 1, both density and likelihood ratio based designs require calculation of the cluster's volume in the input domain $\mathcal{D}$ (e.g., geometric area in two-dimensional Euclidean space). In top-down based enumeration frameworks (e.g., the spatial scan statistic where all candidate regions of a pre-defined parametric shape are exhausted), it is trivial to calculate the volumes or areas (e.g., πr^2 for circular shaped candidates). However, area calculation is not well-defined in the DBSCAN framework, whose output clusters are represented by "maximal point sets".

Furthermore, each test statistic relies on certain normalization to make different clusters comparable [41]. For example, density uses the cluster area as a normalization. One major disadvantage of density is its strong bias toward small clusters [22, 41] (e.g., a cluster of highest density will always

[1]The technique received the Test-of-Time Award at ACM SIGKDD in 2014 due to its impact.

Table 1. Example Candidates of Test Statistics for DBSCAN

Test statistic	**Area of cluster**	**Normalization**	**Bias toward small clusters**	**Computation**
Density d	Required	Area	Yes [22, 41]	Area dependent
Likelihood ratio lr	Required	Area + Null hypothesis	Yes (less) [34, 39, 41]	Area dependent
Cluster size n	N/A	Search context dependent, e.g., fixed radius (or scale), $(\epsilon, minPts)$ in DBSCAN	No	$O(1)$ for a given cluster

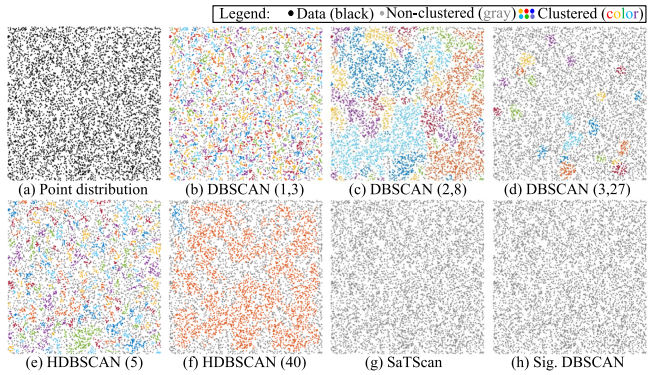


Fig. 1. A 2,000-point realization of the null hypothesis H_0. The parameters in parenthesis are DBSCAN $(\epsilon, minPts)$ and HDBSCAN $(minSize)$.

have the smallest area). To reduce the effect, likelihood ratio additionally incorporates hypothesis-based likelihoods into the normalization. However, as shown by many studies, the design is still biased toward clusters of smallest scales [33, 34, 41, 43], which can be especially problematic for bottom-up style (i.e., local-density-based) clustering algorithms such as DBSCAN.

Cluster size n (i.e., cardinality of a point-set) is another measure used in scan statistic type of methods [41] that does not require the area. To make clusters comparable, it does require some normalizing or constraining conditions to be enforced into the cluster search process; otherwise a bigger set of points is always superior, resulting in non-meaningful outputs. For DBSCAN, the constraining conditions come naturally through the local density criterion θ defined by $(\epsilon, minPts)$. The search radius ϵ and minimum number of points $minPts$ clearly define the conditions that valid cluster points must satisfy, constraining the size of valid clusters (point sets) $n = |C_\theta|$.

In addition, in References [7, 8, 23], one issue discussed is the existence of "small" clusters in the output, which are likely results of natural randomness rather than meaningful clusters. Our preliminary work Significant DBSCAN extends this good start by formalizing the definition of "small" using statistical significance. Previously, to mitigate the "small" cluster issue, a remedy used is to enforce a hard-threshold on minimum cluster size (e.g., default "5" in Reference [1]). While intuitively small clusters (e.g., with only 2 points) are likely to be spurious patterns, spurious patterns are not necessarily small. Figure 1 shows the results of DBSCAN and HDBSCAN on a random point distribution generated by a homogeneous point process. In this point distribution, all clusters detected are chance patterns. Although the chance patterns are indeed small in a few

ALGORITHM 1: Monte Carlo estimation of n_α

Require:
- Total number of points N and spatial domain $\mathcal{D}$
- DBSCAN parameters $(\epsilon, minPts)$
- Significance level α and number of Monte Carlo trials M

```
nList = new List(M)
for i = 1 to M do
    data_r = getRandomPointDistribution(N, D)
    clusters = DBSCAN(data_r, ε, minPts)
    nList(i) = max(clusters.getSizes())
end for
nlist = nlist.sort('DESC')
return n_α = nList(ceil(α · M))
```

results (e.g., Figure 1(b), (d), and (e)), they turn out to be quite large (e.g., thousands of points) in others. Thus, the exact definition of "small" has to depend on many factors, such as the input data, the DBSCAN (or HDBSCAN) criterion θ, the desired significance level and the null hypothesis.

Thus, Significant DBSCAN uses cluster size $n = |C_\theta|$ as the test statistic, and uses significance testing to identify the exact threshold of "small" (i.e., minimum cluster size n_{min}) under all these factors to remove spurious patterns (Figure 1).

3.1.2 Significance Testing for a Single $(\epsilon, minPts)$ Pair. Here we discuss significance testing for a given pair of $(\epsilon, minPts)$ in DBSCAN (the multi-density version is discussed next in Section 3.2). Denote the significance level as α (e.g., 0.01 and 0.05), the size of a detected cluster C as n_C, the total number of points in the point distribution as N, and the domain of the point distribution as $\mathcal{D}$. Further, denote $p_{null}(n_C, N, \mathcal{D}, \epsilon, minPts)$ as the probability of having a cluster of size n_C or greater in a N-point distribution in domain $\mathcal{D}$ generated by a homogeneous point process (i.e., p value). We have the following definition:

Definition 3.1. Cluster C is statistically significant if its p-value $p_{null}(n_C, N, \mathcal{D}, \epsilon, minPts) < \alpha$.

Currently there still does not exist a known statistical model that can calculate the probability p_{null} in a closed-form, as this requires considering the search and expansion process of DBSCAN as well as the randomness associated with distributing N points in an input domain $\mathcal{D}$, which can have irregular boundaries. Thus, we use a Monte Carlo method to estimate p_{null}.

3.1.3 A Baseline Algorithm with Monte Carlo Estimation. In Monte Carlo estimation (Algorithm 1), we generate M simulation trials to approximate the distribution of cluster size n (i.e., the test statistic) in point distributions generated by a homogeneous point process. In each trial, we first generate a random N point distribution using the homogeneous point process in domain $\mathcal{D}$, and then run DBSCAN with the same input $(\epsilon, minPts)$ to get the best or maximum cluster size $\hat{n}$ in the trial. After M trials, we will have M best $\hat{n}$ values. By sorting the M values in descending order, we can estimate the p-value p_{null} of a cluster C detected from the real data by checking its rank r in the sorted list: $p_{null}(n_C, N, \mathcal{D}, \epsilon, minPts) = r/M$. Note that M has to be at least $1/\alpha$ to evaluate the significance.

We reject the null hypothesis and mark cluster C as significant if $p_{null} < \alpha$ (or $r < \alpha M$). Equivalently, we just need to find the (αM)th largest value in the sorted list and use that as a threshold (denoted as n_α) of cluster size to filter out non-significant clusters.

3.1.4 Summary of a Dual-convergence Algorithm. To speed up the additional computation introduced by the Monte Carlo estimation, we proposed a dual-convergence algorithm to reduce

unnecessary executions of the exact DBSCAN sub-routine across Monte Carlo trials. To avoid redundancy, here we will summarize the three key points of the algorithm (details in Reference [42]):

- **Upper-bound for significant clusters:** We use a discrete version of DBSCAN, where exact point distribution is discretized into a grid, and original ϵ-based density criteria are projected onto the discrete space. The upper-bounding search neighborhood (a sub-grid) is then the union of all possible ϵ-neighborhoods with a center inside the center cell of the sub-grid. Further, using integral image, the number of points in the sub-grid can be calculated in $O(1)$ time, greatly reducing the cost. If the upper bound cluster size returned by the discrete scan is smaller than the detected cluster candidate, then the exact DBSCAN is skipped.
- **Early-termination for spurious clusters:** Since rejecting a spurious cluster only requires $100\alpha\%$ of trials with a better test statistic value, the rest of the trials can be skipped as soon as the condition is met. Our probability analysis [42] shows that early-termination is highly effective for reducing computational time on spurious clusters (e.g., the probability of terminating at the 20th trial, of 1000, is 0.52 for $\alpha = 0.01$).
- **Dual-convergence coordination:** Since the list of DBSCAN clusters (e.g., tens or hundreds as shown in Figure 1) that need significance testing is a mixture of significant and spurious clusters, neither of the two techniques discussed above works well alone. Dual-convergence provides coordination between the two components and manages the cluster list to gradually improve speed-up as more trials are executed.

3.2 Finding Clusters of Varying Densities with Uniform Sampling

Using the significance testing framework, the single density criterion based DBSCAN can be easily extended to identify clusters of varying densities. This is made possible by robust elimination of spurious results at each density level.

Denote $S_\theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$ as a set of density criteria for DBSCAN. Further, denote $S_{all} = S_{true} \cup S_{spurious}$ as the set of all clusters detected by all criteria in S_θ, where $S_{true} = \{S_{true}^{\theta_i} | \forall i = 1, \ldots, k\}$ and $S_{spurious} = \{S_{spurious}^{\theta_i} | \forall i = 1, \ldots, k\}$ are the sets of all true clusters and spurious clusters detected using each $\theta_i \in S_\theta$ for an input dataset, respectively. Using the original DBSCAN, even if all the clusters S_{all} across different densities can be obtained by separately executing DBSCAN with each density criterion in S_θ, it is difficult to merge all the results in S_{all} due to two main reasons: (1) There can be heavy overlaps between sets of clusters $S_{all}^{\theta_i}$ detected from different density criteria θ_i, and it is hard to select which subset to keep, and (2) aggregating results S_{all} across densities merges not only true clusters $S_{true}^{\theta_i}$ of each density θ_i but also spurious clusters $S_{spurious}^{\theta_i}$, thereby substantially increasing the number of spurious patterns detected. As we will show in the following, the significance modeling offers an effective way to greatly mitigate these two issues. The multi-density framework in Significant DBSCAN has two key components:

- **Density sampling:** During the initialization, the minimum and maximum density θ_{min} and θ_{max} are estimated from the input data in $\mathcal{D}$. Then, uniform sampling is used to generate k density criteria to construct a list S_θ, where θ values in S_θ are sorted in descending order.
- **Sequential θ-feeding and removal:** Multi-density detection in Significant DBSCAN starts with the highest density criterion θ_{max} in S_θ due to the one-way directionality of density in DBSCAN, i.e., a higher density criterion is not satisfied by lower density clusters. Thus, starting from θ_{max} helps avoid identifying clusters at other density levels. After all clusters $S_{all}^{\theta_i}$ are detected using a criterion θ_i, spurious results $S_{spurious}^{\theta_i}$ are filtered out through significance testing and the rest are pushed into the final output set. To avoid these clusters being re-detected at a lower density level, the associated points of the clusters are removed from the data before moving into the next density criterion.

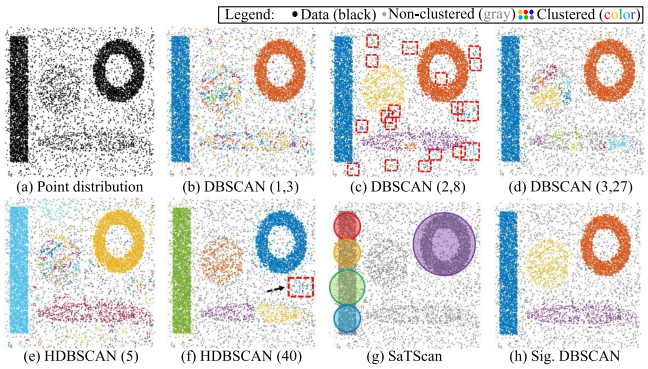


Fig. 2. Example result on clustered data generated by H_1 (noise: gray).

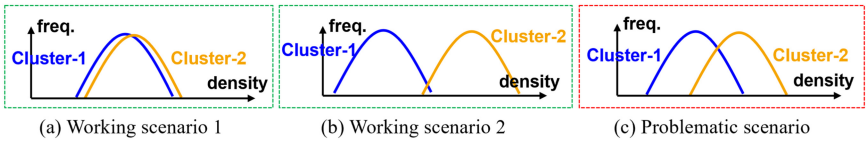


Fig. 3. Limitations of our previous work on Significant DBSCAN.

As we can see, Significant DBSCAN, combined with the sequential high-to-low search strategy, can effectively mitigate the issue of overlapping clusters across different θ, and robustly remove spurious patterns in the aggregated results (full details in Reference [42]). Finally, Figure 2 shows an example result of Significant DBSCAN compared with other approaches, i.e., DBSCAN, HDBSCAN [1, 7, 8, 23] and the spatial scan statistic (SaTScan [3, 17]). In the experiments, we varied parameters of other approaches (e.g., DBSCAN, HDBSCAN) while the parameters for Significant DBSCAN (i.e., the number of density criteria k and significance level α) are kept the same across all the datasets.

3.3 Limitations

In order for Significant DBSCAN to perform well in this multi-density framework, two conditions need to be met. First, an estimated density (i.e., one of the k uniform samples in $[\theta_{min}, \theta_{max}]$) needs to be from a working range (e.g., not too high so that a substantial portion of the true cluster is missed) to approximately capture a true cluster at the closest density level. Second, the densities of the true clusters are either very similar or very different. Denote C_1 and C_2 as two true clusters having densities θ_1 and θ_2, respectively. Since a point process—no matter following H_0 or H_1—is a random process, the exact densities across local neighborhoods will not be identical; instead, they follow a random distribution. Figure 3 shows several illustrative scenarios of local density distributions of true clusters C_1 and C_2. The X axis represents the density of a local neighborhood, and the Y axis represents the frequency of local neighborhoods having a density value on X. As shown in Figure 3(a), when their density distributions (independent from spatial distributions) are very similar, a working density criterion θ for C_1 can also well capture C_2, so the approach remains working well. Similarly, in Figure 3(b), the density distributions are well separated so a working density criterion θ for C_2 will not falsely detect a significant proportion of C_1. However, when there exists a moderate overlap between the density distributions (Figure 3(c)), there is a high risk that a lower density cluster (e.g., C_1) will be shattered in the final output. Examples can also be found in Figure 2(c) and (d).

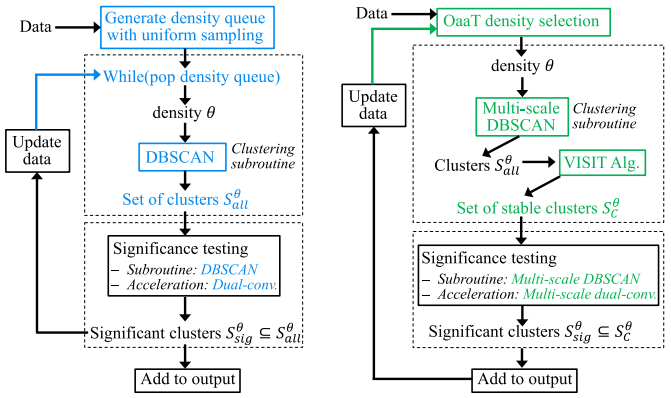


Fig. 4. Overall frameworks of Significant DBSCAN (previous work) and Significant DBSCAN+ (extension).

4 SIGNIFICANT DBSCAN+

4.1 Overall Framework

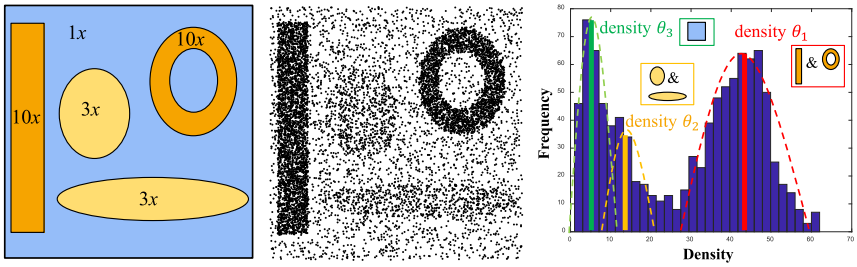
Figure 4(b) shows the overall framework of Significant DBSCAN+, which is an extension of our previous work to address its limitations. Comparing the architectures in Figure 4(a) and Figure 4(b), there are three major changes in Significant DBSCAN+:

- First, we propose two strategies that work collaboratively to improve the chance of selecting a working density criterion θ to capture the true clusters:
  (1) Instead of the rigid uniform-sampling based density selection, we employ a **One-at-a-Time (OaaT)** density selection approach to recursively identify peak-densities in an input dataset, reducing the confusion caused by overlapping densities;
  (2) We propose a multi-scale DBSCAN sub-routine to expand the range of density criteria θ that can be used by the sub-routine to correctly capture the true clusters at each density level.
- Second, we propose a VISIT algorithm to improve the separation among clustered point sets belonging to different density levels, and thus further mitigate the shattered-cluster problem when the density-distribution of clusters has moderate overlaps (Figure 3(c)).

4.2 One-at-a-time Density Selection

Density selection is an important step in density-based clustering frameworks such as DBSCAN. Given a neighborhood size ϵ, Figure 5 shows the distribution of densities (i.e., number of points in ϵ) of a 10,000-point dataset generated using a clustered point process H_1. The distribution contains four true clusters C_1 to C_4. Denote x as the background probability density (i.e., outside the four clusters in $\mathcal{D}$). The inner-space of clusters C_1 and C_2 (the rectangle and ring) both have a probability density of $10x$, and the inner-space of C_3 and C_4 (the circle and ellipse) have $3x$. The expected density values of the clusters and the background are represented by the vertical lines in Figure 5(c).

As discussed in Section 3, our preliminary Significant DBSCAN uses a uniform sampling approach to select k densities from $[\theta_{max}, \theta_{min})$, which has limited flexibility and may cause clusters to shatter into statistically-significant pieces. There are two main challenges in directly estimating the true expected densities from data. First, the number of distinct probability densities (e.g., $10x$, $3x$ and $1x$ in this case) is unknown, making it difficult to robustly extract all the expected true den-



(a) Point process (b) Realization of a point process (c) Histogram of density samples

Fig. 5. Illustrative example of distribution of densities. For visualization purposes, true densities θ_1 to θ_3 are calculated directly using the probability densities and corresponding volumes shown in (a), which are unknown in practice (i.e., can only observe realizations). Dashed in (c) lines are hand-drawn approximations.

sities $\{E_{den}\}$. Second, there can be heavy overlaps between density distributions (e.g., lower-tail of $10x$ and upper-tails of $3x$ and x), increasing the uncertainty and hardness of separating out true densities being mixed in the middle. As the goal of this sub-task is not to address this classical mixture model problem but to help improve subsequent detection of significant clusters, we propose a OaaT density selection strategy to reduce the difficulty of this sub-task by leveraging the advantage of significance testing.

To avoid the need to know the number of distinct density groups in the mixed distribution, the OaaT strategy only attempts to identify the expected true density E_{den} associated with the highest probability density x_{max} in the current dataset. This leverages the fact that the general Significant DBSCAN+ framework only needs to detect clusters belonging to one density group at a time. Moreover, the density distribution corresponding to x_{max} is on the extreme side of the overall distribution, so the peak frequency associated with its expected true density E_{den} is not affected by tails of distributions from two sides, making it substantially easier to separate out. Finally, as Significant DBSCAN+ by design removes points belonging to a significant cluster after completing each density level, the effect of the density values associated with the current x_{max} will also be taken out for the next round of OaaT density selection. This allows the OaaT strategy to remain effective during its recursive application through the rounds.

Using the OaaT strategy, the density selection algorithm operates as follows. First, given an ϵ, density values—represented by the number of points in ϵ neighborhoods—are sampled and converted into a density-frequency histogram (e.g., Figure 5(c)) with h bins (h is defaulted to 30, and the sensitivity will be evaluated in Section 5). As the expected true density E_{den} corresponding to the highest probability density x_{max} is represented by a frequency peak (i.e., vertical line on the right in Figure 5), a typical local maximum detector (window size defaulted to 5) is used to extract the peak and its density value on the X axis (e.g., θ_1 in Figure 5).

4.3 A Multi-scale DBSCAN Sub-routine

Selecting a proper density to construct a density criterion θ is only one important step of the problem. To capture the desired clusters, we also need to make sure that θ is effective for the DBSCAN sub-routine, especially considering DBSCAN's (and many other density-based clustering approaches') sensitivity to the input density criterion.

Figure 6 shows an example of the sensitivity, where the dataset is a 10,000-point distribution in a 100×100 area. Similarly, the rectangle and the ring have a probability density of $10x$ for generating a point at each location; the circle and the ellipse have $3x$; and the background has $1x$. Using $\epsilon = 2$, the true density (i.e., number of points in a circular neighborhood with a radius of 2)

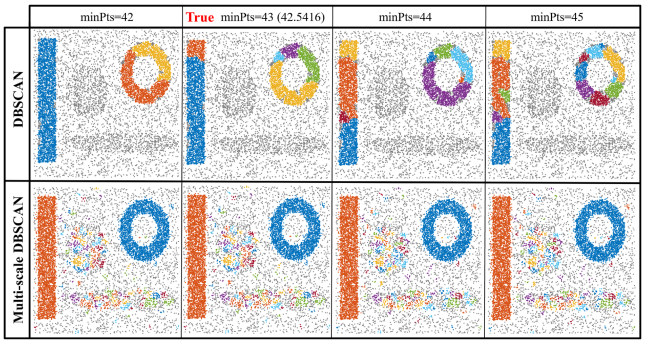


Fig. 6. Sensitivity of DBSCAN to density parameters. The second column is the true density directly calculated for the rectangle and ring using the point process (not observable in practice).

for the rectangle and the ring is about 42.5416. As shown in Figure 6, a DBSCAN density criterion $\theta = (\epsilon = 2, minPts = 42.5416)$ constructed using the known true density is not able to correctly capture the corresponding true clusters. In other words, the true density is not within the working range of the density criterion for DBSCAN, because the contiguity of the true density at ϵ is broken by random effects inside the true clusters. In addition, by only varying the *minPts* parameter in the density criterion θ by a minimum step (i.e., a single point), the clustering results become very different as shown in Figure 6. This sensitivity of DBSCAN makes it difficult to construct a working density criterion using the estimate selected by the OaaT strategy.

To address this sensitivity issue, we propose a multi-scale DBSCAN sub-routine to replace the DBSCAN sub-routine used in our preliminary work. The key change in multi-scale DBSCAN is that the validation of the local density criterion is extended from a single scale to multiple scales. In DBSCAN, a point passes the local criterion $(\epsilon, minPts)$ if there is at least *minPts* points inside its local ϵ neighborhood. Denote ϵ_{min} and ϵ_{max} as the minimum and maximum neighborhood sizes, the multi-scale DBSCAN additionally projects the original criterion $\theta_{ori} = (\epsilon_{ori}, minPts_{ori})$ onto k scales in the range $[\epsilon_{min}, \epsilon_{max}]$, and the new multi-scale criterion becomes

$$\theta_{multi} = \{\theta_i = (\epsilon_i, minPts_i) | \, minPts_i = minPts_{ori} \cdot \frac{\epsilon_i^2}{\epsilon_{ori}^2}, \forall i = 1, \ldots, k\} \cup \theta_{ori}.$$

Further, denote SAT_θ as a Boolean variable that represents if a point satisfies the criterion θ. In DBSCAN, a point is a core point if $SAT_{\theta_{ori}} = 1$, and all points inside ϵ_{ori} are added to the expansion list. In the multi-scale extension, the new rule is defined as follows.

Definition 4.1 (Multi-scale Local Rule). The rule has two components: (R1) Core point rule: A point is a core point in multi-scale DBSCAN if $SAT_{multi} = 1$, where

$$SAT_{multi} = SAT_{\theta_1} \vee SAT_{\theta_2} \vee \ldots SAT_{\theta_k} \vee SAT_{\theta_{ori}};$$

and (R2) Expansion rule: All points within ϵ_{ori} are added to the expansion list if $SAT_{multi} = 1$.

As shown in Definition 4.1, this multi-scale test SAT_{multi} is only used to determine if a point is a core point, and $SAT_{multi} = 1$ as long as a criterion θ_i is satisfied at any scale. After the test, the expansion rule remains the same as the single scale DBSCAN approach. This definition guarantees that the new rule is strictly less harsh than the single-scale rule in DBSCAN. While the random effects may break the density contiguity of true clusters at a single scale ϵ (Figure 6), the new test across multiple scales aims to suppress such effects by sampling more local neighborhoods at each point.

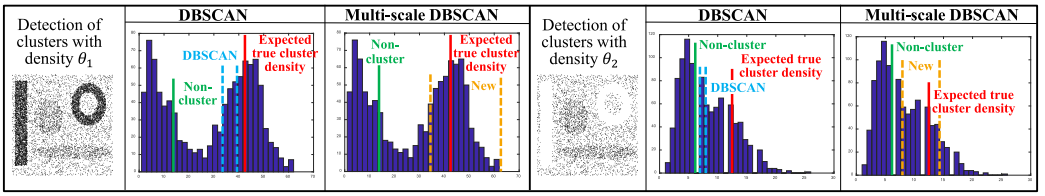


Fig. 7. Visualizing the expansion of working density ranges with multi-scale DBSCAN.

Figure 6 (bottom row) shows the clustering results of the multi-scale DBSCAN sub-routine using the same set of density criteria. The number of scales k is defaulted to 11 and its effect will be evaluated in Section 5. As we can see, the multi-scale extension is able to correctly capture the corresponding true clusters using the true density (i.e., $\theta_{ori} = (\epsilon = 2, minPts = 42.5416)$) as well as the other nearby higher or lower values. Note that a side-effect of the multi-scale rule is that some spurious results corresponding to other lower-density clusters are captured at the same time. While these newly introduced spurious clusters will be problematic in a regular scenario, the incorporation of statistical significance allows convenient elimination of these undesired by-products. In addition, another dedicated new design—stable subset selection and testing—will be introduced in Section 4.4 to robustly avoid inclusion of these undesired pieces. In this section, the discussion will focus on capturing the true clusters associated with the current density level.

To better visualize the range of working densities for capturing the true clusters in Figure 6, we enumerated through a list of different density criteria with $\epsilon_{ori} = 2$. The candidates for $minPts$ are enumerated by gradually expanding around the expected true density 42.5416. In the visualization shown in Figure 7, a density is considered as a working density if each of the two corresponding clusters (i.e., the rectangle and the ring) is captured by a single contiguous cluster that has no more than 5% of missing or extra points. The working ranges of densities of DBSCAN and the multi-scale DBSCAN are shown by the dashed blue and orange lines in Figure 7, respectively. The solid red line represents the expected true density E_{den} of the two clusters associated with the highest probability density. In contrast, the green line shows the non-cluster density $E_{den}^{H_0}$, which is the expected number of points in a $\epsilon = 2$ neighborhood assuming the 10,000-point data is generated by a homogeneous point process H_0 (i.e., data are purely noise and have no true cluster). Thus, any value below $E_{den}^{H_0}$ is no longer meaningful for cluster detection. As we can see, the expected true density E_{den} is outside DBSCAN's the working range of density, which can also be confirmed by the clustering results shown in Figure 6. In contrast, the working range of the multi-scale DBSCAN contains E_{den} and thus can correctly capture the true clusters with it. In addition, the other important observation is that the working range of the multi-scale DBSCAN has greatly expanded compared to that of the DBSCAN, reducing its sensitivity to an input density.

The same trend can be seen in Figure 7 (right), which shows the working ranges of the two sub-routines for the next density level associated with the circle and the ellipse. As discussed in Section 4.2, significant clusters are removed from the data after each round of the detection, which helps remove their effects on the density distribution. As we can see, the working range of density for DBSCAN here is not only very narrow but also very close to $E_{den}^{H_0}$, making it difficult to capture the true clusters. In comparison, the working range for the multi-scale DBSCAN again contains the expected true density and is much less sensitive with the wider coverage.

From a computational perspective, however, the use of multiple scales would subsequently lead to additional computational overhead. This will be addressed in Section 4.5.

4.4 Stable Cluster Set Selection and Testing

Finally, to mitigate the cluster shattering problem caused by density overlaps (Section 3.3), we aim to reduce the chance of undesired inclusion of the sub-portions of lower density clusters in the clustering results. Specifically, denote $S_C^\theta = \{C_1^\theta, C_2^\theta, \ldots\}$ as a set of clusters corresponding to the current density level θ (i.e., highest density selected by OaaT in Section 4.2), and $S_{seg}^\theta = \{seg_1^\theta, seg_2^\theta, \ldots\}$ as a set of sub-segments from lower density clusters (or random noises) being detected due to the overlap in density distributions (Figure 3(c)). As the result returned by OaaT and multi-scale DBSCAN (e.g., Figure 6) is a combined set of both (i.e., $S_{all}^\theta = S_C^\theta \cup S_{seg}^\theta$), the goal is to identify clusters from $S_C^\theta \subseteq S_{all}^\theta$ and only use them for significance testing. In other words, we need to exclude sub-segments of lower density clusters in S_{seg}^θ from the result of the current round. We call the desired set S_C^θ as a set of stable clusters (i.e., not sub-segments).

We propose a **Virtual sequence visit (VISIT)** algorithm to select the set of stable clusters. The main idea, of the VISIT algorithm is that, when multiple stable clusters exist in S_C^θ, each cluster $C_i^\theta \in S_C^\theta$ tends to maintain its size in the detection result no matter if the other clusters in S_C^θ are removed from the data, whereas clusters in S_{seg}^θ tend to have major changes in their sizes after S_C^θ is removed. This is because clusters in S_C^θ are at the same density level θ, so approximately the same density will be returned by the OaaT strategy as long as one of them is still in the data. In contrast, when all clusters in S_C^θ are removed, a major change $\Delta\theta$ will result in the density selected by OaaT and this change will subsequently cause clusters in S_{seg}^θ to grow and merge (e.g., segments belonging to the $(\theta - \Delta\theta)$ density group will grow to stable clusters).

Leveraging this characteristic, after the initial round of detection at the current density level θ, the VISIT algorithm keeps a sequence $seq = C_1, C_2, \ldots$ of all detected clusters sorted in descending order by their test statistic values (i.e., cluster sizes as defined in Section 3.1.1). Then, VISIT creates a virtual copy of the data, and starting from the first cluster C_1 in the sequence seq, it removes the cluster from the data and re-runs OaaT and multi-scale DBSCAN. If the best cluster returned from the virtual data does not differ from the second cluster C_2 in seq (i.e., the next best cluster after C_1) by a small tolerance $\delta = 5\%$, then C_2 will be considered as a stable cluster. The VISIT algorithm then further removes C_2 from the virtual data and repeats the same process until the tolerance δ is violated. The removed clusters form the set of stable clusters as illustrated in Algorithm 2.

Finally, after a set of stable clusters is identified, we perform significance testing via the Dual-Convergence algorithm described in Section 3.1.4. If there exists at least one significant cluster, then we will remove the significant clusters from the data and start another round of the VISIT algorithm; otherwise, Significant DBSCAN+ will terminate and output existing clusters as the final output.

4.4.1 Note on Multiple Testing. A common concern for performing significance testing on multiple patterns is the issue of multiple testing, i.e., the probability of falsely rejecting a null hypothesis H_0 (i.e., the type-I error) is amplified to be greater than the significance level α. For example, if the tests are all performed independently on t independent patterns, then the type-I error becomes $1 - (1 - \alpha)^t > \alpha$. Although Significant DBSCAN+ often returns multiple clusters, the approach is in fact not susceptible to the issue of multiple testing. Specifically, given a dataset generated by H_0, Significant DBSCAN+ guarantees that it returns an empty output with a probability greater than $1 - \alpha$. This is achieved by the design that only the best cluster from each Monte Carlo simulation trial is used for p-value estimation (Algorithm 1, Section 3).[2] As a result, the best cluster detected in a dataset following H_0 has strictly less than α probability of passing the significance testing

[2]Similar design strategies are employed in scan statistic methods to avoid multiple testing issues.

ALGORITHM 2: The VISIT Algorithm

Require:
- Input data X in domain $\mathcal{D}$
- Base parameters for OaaT and the multi-scale DBSCAN: (base neighborhood size: ϵ, scale range: $[\epsilon_{min}, \epsilon_{max}]$)
- A tolerance for checking the stability of a cluster: δ

```
stable = initEmptyClusterList()
{Initialization: Get a sequence of clusters}
den = selectDensityByOaaT(X, ϵ)
Θ = getParametersForMultiDBSCAN(den, ϵ, [ϵ_min, ϵ_max])
seq = MultiDBSCAN(X, Θ)
seq.sort('DESC', size)
{Select a set of stable clusters}
X' = copy(X)
while TRUE do
    if stable.isEmpty() == TRUE then
        C = seq.pop()
        stable.add(C)
        X'.remove(C)
    else
        den' = selectDensityByOaaT(X', ϵ)
        Θ' = getParametersForMultiDBSCAN(den', ϵ, [ϵ_min, ϵ_max])
        C' = getBest(MultiDBSCAN(X', Θ'))
        C = seq.pop()
        if setDiff(C,C') ≤ δ then
            stable.add(C)
            X'.remove(C)
        else
            break
        end if
    end if
end while
return stable
```

(i.e., it has to be in the top $100\alpha\%$ of the simulated distribution). In addition, as discussed in Section 4.4, a significant cluster has to exist before the search can continue onto the next round of detection. Such dependency between consecutive rounds also avoids the multiple testing issue.

4.5 Computational Consideration

In this section, we discuss the computational aspects of Significant DBSCAN+ with two algorithmic improvements to handle the additional search cost of the multi-scale DBSCAN.

Compared to Significant DBSCAN (our previous work in Section 3), the use of the new multi-scale DBSCAN sub-routine requires extra computation as the density criteria θ_{multi} need to be tested on all k scales as defined in the multi-scale rule (Definition 4.1). Thus, here we develop algorithmic improvements to reduce the extra cost. In addition, as Significant DBSCAN uses the DBSCAN sub-routine both in exact and discrete forms (i.e., for upper bound computation in Section 3.1.4), we present two acceleration methods, i.e., a recursive search for the exact version in Section 4.5.1, and an equivalence-class compression approach for the discrete form in Section 4.5.3.

4.5.1 Recursive Search. To reduce the cost of checking against the multi-scale density criteria θ_{multi}, we use a scale-recursive structure of θ_{multi} to gradually confine the search scope, where the criteria in θ_{multi} are sorted by ϵ (equivalently, $minPts$) in a descending order. The range query on the original data is then only executed once at the beginning for each data point using the maximum ϵ_1 in θ_{multi}. Then, denote NB_i and NB_{i+1} as the returned neighborhood point sets for

range queries using ϵ_i and ϵ_{i+1}, respectively. Since $\epsilon_i > \epsilon_{i+1}$, we always have $NB_{i+1} \subseteq NB_i$. Thus, for the range query at scale ϵ_i ($i > 1$), the scale-recursive search can inherit the previous result NB_{i-1} and only perform the query on NB_{i-1} to obtain the same result.

4.5.2 Equivalence Class Compression. For the discrete version of DBSCAN used in the dual-convergence algorithm (Section 4.5), the exact point distribution is no longer used or returned by a range query. Thus, the recursive algorithm is not applicable here; in fact, it will result in a higher cost if used (compared to integral image based calculation). To reduce the cost, we develop an equivalence-class compression strategy to merge criteria of nearby scales into an equivalent group. At a high level, the idea, is that the set of multi-scale criteria in the original continuous space can be reduced to a smaller set of criteria in the discrete space due to the effect of "binning" during discretization. Similar to the previous section, denote $\theta_{multi} = \{\theta_i = (\epsilon_i, minPts_i) | \, i = 1, \ldots, k+1)\}$ as the set of multi-scale criteria where $\epsilon_i > \epsilon_{i+1}$ (i.e., descending order). Further, denote g_i as the above-mentioned uncertainty-bounding sub-grids of ϵ_i; here a sub-grid always has an odd number of cells on each edge, and the center cell contains the point from which ϵ_i is measured in the exact version. The equivalence-class compressed criteria is then defined as follows.

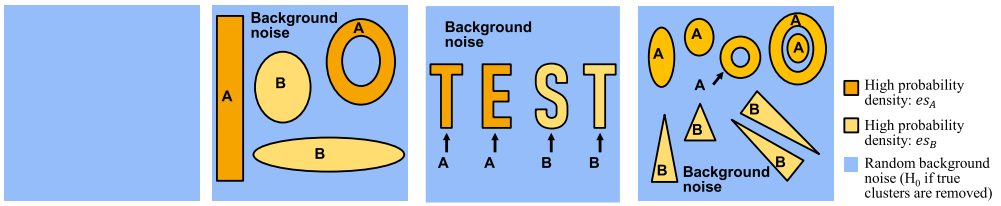
Definition 4.2 (Equivalence Class Compressed Criteria). An equivalence class EC refers a set of neighboring multi-scale criteria where different ϵ_i correspond to the same g due to space discretization. Given the original set of criteria in the discrete space $\theta^{dis}_{multi} = \{(g_i, minPts_i) | \, i = 1, \ldots, k+1\}$, the compressed set of criteria $\theta^{ec}_{multi} = \{(g_i, minPts_i) | \, g_i = g_{i-1}$ and $g_i \supset g_{i+1}$ where $i \in [2, k]$; **or** g_i where $i = k+1$; **or** $g_i \supset g_{i+1}$ where $i = 1\}$, where the minimum $minPts_i$ is selected for the criteria for each equivalence class EC_j, as it must be satisfied if any $minPts_i \in EC_j$ is satisfied.

4.5.3 Time Complexity. Denote N as the data size, k as the number of scales used in the Multi-scale DBSCAN, M as the number of Monte Carlo trials, and $|G|$ as the number of grid cells in the discrete version of DBSCAN. Further, denote ρ as the proportion of trials that require execution of the exact algorithm (i.e., when bounds do not work), q as the average number of points inside the maximum ϵ-neighborhood (used as a bound for range query complexity in recursive search after the first scale) around the N data points, and γ as the compression ratio achieved by the Equivalence Class compression. For DBSCAN, as recently corrected in Reference [13], its worst-case complexity is $O(N^2)$ and the $O(N \log N)$ algorithm so far only exists for data with a dimension ≤ 2. Thus, here we use $O(N^2)$ for a more general analysis. The complexity of the baseline algorithm (including dual-convergence acceleration inherited from Significant DBSCAN) is then $O(\rho M \cdot kN^2 + (1-\rho)M \cdot k \cdot (N + |G|))$, and the complexity of the accelerated algorithm is bounded by $O(\rho M \cdot (N^2 + (k-1)Nq) + (1-\rho)M \cdot (\gamma k) \cdot (N + |G|))$. As the discrete version (used to create bounds, Section 3.1.4) runs in linear time $O(|G|)$, $O(N + |G|)$ just adds the $O(N)$ time needed to aggregate points to grid cells. In practice, a discrete scan is used when $|G| < N^2$. Also, the complexity results are for one density criterion selected by OaaT (so N is for the current data size in a round), and they can be applied to each distinct density identified by the algorithm. Finally, for non-clustered data, the algorithm often terminates very early (Section 3.1.4) after a small proportion λ of trials (e.g., λ may be just slightly greater than the significance level α). As the cluster sizes from non-clustered data often cannot surpass the upper-bounds from discrete scan, we can ignore ρ (e.g., $\rho \approx 1$) in the results above and the complexity becomes $O(\lambda M \cdot (N^2 + (k-1)Nq))$.

5 VALIDATION

Our experiments aim to answer the following questions:

- Are the candidate methods susceptible of finding spurious patterns? Is Significant DBSCAN+ robust against such spurious results?



(a) H_0 (no true cluster) (b) "Shape" process (c) "Test" process (d) "Dense" process

Fig. 8. Different base landscapes for generating point processes under various conditions specified by data size N and effect sizes es. Hotspots in the "shape" process in (b) have relatively larger footprints, leading to stronger signals of clusters. In contrast, hotspots in the "test" process in (c) are much smaller and more difficult to detect. Finally, the "dense" process in (d) contains a denser distribution of small hotspots (i.e., 9) and has a smaller noise-to-hotspot ratio compared to the "test" process.

- Does the extended version, Significant DBSCAN+, improve solution quality over Significant DBSCAN, especially when cluster densities have moderate overlaps (Section 3.3)? More broadly, does Significant DBSCAN+ improve solution quality over other candidate methods?
- Do the computational enhancements reduce time cost for data following both H_0 and H_1?
- How does the performance of Significant DBSCAN+ change with varying parameters?

5.1 Solution Quality

5.1.1 Data Generation. To evaluate the solution quality, we use statistical definitions from Section 2 to generate datasets where ground-truth clusters can be inserted into the point processes and recorded to compute performance statistics [19, 22, 41]. As shown in Figure 8, datasets generated under H_0 (homogeneous point processes) have identical probability densities across the whole space, which means there is no true cluster and any detections from datasets generated by H_0 are spurious results. In contrast, true clusters (i.e., highlighted by orange-colored regions) in clustered point processes H_1 have higher probability densities of generating points than the outside.

Here we generate 5,400 datasets under a variety of conditions to quantitatively evaluate the solution quality: (1) Varying total number of points $N \in \{2{,}000, 4{,}000, 6{,}000, 8{,}000, 10{,}000, 12{,}000\}$; (2) Varying hypotheses H_0 and H_1; and (3) For H_1, varying effect size $es = (es_A, es_B)$, where each effect size value represents how many times the probability density pd_{in} inside a corresponding true cluster is as high as that of the outside pd_{out} (i.e., background colored in blue in Figure 8); es_A and es_B are the effect sizes for the true clusters marked with A and B in Figure 8, respectively. The effect size $es \in \{(2,4), (3,6), (4,6), (4,8), (6,8), (3,12)\}$. The values for N and es are defaulted to 10,000 and $(4,6)$ when used as controlled parameters. Each parameter setting is further combined with the three different base landscapes (i.e., "shape," "test," and "dense") shown in Figure 8(b), (c), and (d). Finally, to account for the natural randomness in point processes, we generate 100 datasets for each combination of parameter setting and base landscape, resulting in 5,400 different datasets in total.

5.1.2 Candidate Methods. Based on the results from our preliminary work, k-means++ [5] and expectation-maximization for Gaussian Mixture Models [32] could not generate reasonable results due to the irregularly shaped clusters. Thus, we used spectral clustering [24, 38] as the partitioning-based approach as it is less sensitive to the shapes. Similarly, since SaTScan also struggled with irregular shapes and multiple-cluster scenarios (Figure 2), we skipped it in this comparison, and instead, added our preliminary work, Significant DBSCAN, as a more suitable baseline for this extension. In addition, we varied the parameters for competing methods (e.g., DBSCAN, Spectral clustering, HDBSCAN) but kept the parameters *exactly the same* for Significant DBSCAN and

Significant DBSCAN+, respectively, throughout detection across all 5,400 datasets under different conditions. The following is a summary of the candidate methods:

- **DBSCAN (two versions)**: We constructed two candidate methods using DBSCAN (DB_A and DB_B) whose density criteria are directly calculated using the ground-truth effect sizes es_A and es_B in each dataset to help improve the methods' solution quality.
- **Spectral clustering**: We directly feed in the ground-truth number of clusters to spectral clustering [24, 38], reducing that extra layer of challenge for it (for data under H_0, we fixed the number to 4, which is the smallest number of clusters used for other datasets). The Laplacian matrix is normalized with random walk (i.e., $L_{norm} = D^{-1}L$, where D is the degree matrix; L and L_{norm} are the original and normalized Laplacian matrices, respectively).
- **Deep embedding clustering**: A widely adopted state-of-the-art deep clustering approach [40]. Again, we directly feed in the ground-truth number of clusters, which is an input to the network. We use the Keras implementation provided in Reference [4].
- **HDBSCAN (three versions)**: A recent state-of-the-art hierarchical density-based clustering approach proposed by researchers including creators of DBSCAN and OPTICS as a powerful integration of the key ideas [1, 7, 8, 23]. HDBSCAN requires a minimum cluster size min_{size} as input (defaulted to 5 in the standard library) to generate the clustering hierarchy. Thus, we constructed three candidate methods—HDB_5, HDB_{100}, and HDB_{200}—with the min_{size} set to 5, 100, and 200, respectively (sizes of all true clusters in experiments are greater than 100).
- **Significant DBSCAN**: Our preliminary work in Reference [42]. Significance level α is 0.01. All parameters are kept exactly the same throughout experiments across all 5,400 datasets.
- **Significant DBSCAN+**: The extended version described in this article. Significance level α is 0.01. All parameters are kept exactly the same throughout experiments across all 5,400 datasets.

5.1.3 Measures. For datasets following H_0, i.e., where no true cluster exists, we use the number of spurious results κ detected to evaluate the robustness of candidate methods. The number is measured at both cluster and data levels: (1) Cluster level: $\kappa_{cluster}$ represents the total number of detected spurious clusters across a set S_{data} of input datasets, and (2) κ_{data} is the number of unique datasets in S_{data} where one or more spurious clusters are detected.

For datasets following H_1 that contain true clusters, we use F1 scores (i.e., harmonic mean of precision and recall) to measure the solution quality. To compute precision and recall rates, we need an additional step to determine which true clusters are successfully detected. Denote $S_C = \{C_1, C_2, \ldots, C_z\}$ as the set of true clusters, and $S'_C = \{C'_1, C'_2, \ldots, C'_{z'}\}$ as the set of detected clusters. As each cluster is essentially a point-set, we use a point-set-based Intersection-over-Union (set-IoU) to determine the success of a detection. Specifically, a cluster $C_i \in S_C$ is successfully detected if $\exists C'_j \in S'_C$, such that: $\frac{C_i \cap C'_j}{C_i \cup C'_j} \geq IoU_{thrd}$, where the threshold IoU_{thrd} is set to 0.8. In addition, we add the traditional clustering measures—**unsupervised clustering accuracy (ACC)** and **normalized mutual information (NMI)**—to the evaluation of candidate methods. The difference between F1 score and ACC (or NMI) is analogous to the difference between object and pixel level measures in object detection and segmentation.

5.1.4 Comparative Analysis. For all the following results in comparative analysis, the parameters of Significant DBSCAN and Significant DBSCAN+ are kept the same as described in Section 5.1.2.

Robustness against spurious results: Table 2 summarizes the results of the candidate methods on datasets following H_0 where no true cluster exists. The value in each cell of the table

Table 2. Number of Spurious Results Detected from Data Following H_0 (Cell Format: $\kappa_{cluster}$ (κ_{data}))

N	DB_A	DB_B	Spectral	Deep	HDB_5	HDB_{100}	HDB_{200}	SigDB	SigDB+
2,000	7,881 (100)	1,371 (100)	400 (100)	396 (100)	7,518 (100)	121 (58)	0 (0)	3 (3)	1 (1)
4,000	4,765 (100)	87 (62)	400 (100)	391 (100)	15,085 (100)	247 (99)	90 (45)	2 (2)	0 (0)
6,000	3,831 (100)	6 (6)	400 (100)	362 (100)	23,116 (100)	298 (100)	174 (81)	0 (0)	1 (1)
8,000	2,027 (100)	0 (0)	400 (100)	353 (100)	29,538 (100)	341 (100)	230 (98)	1 (1)	1 (1)
10,000	1,496 (100)	0 (0)	400 (100)	345 (100)	37,754 (100)	369 (100)	252 (98)	0 (0)	2 (2)
16,000	218 (91)	0 (0)	400 (100)	364 (100)	58,637 (100)	491 (100)	324 (100)	1 (1)	0 (0)

represents the total number of spurious patterns detected, $\kappa_{cluster}$, across 100 datasets generated using the corresponding number of points N; the value in parentheses is the corresponding data-level κ_{data}, i.e., the number of datasets where one or more spurious clusters are returned (Section 5.1.3). As we can see, all other methods (i.e., different versions of DBSCAN and HDBSCAN, as well as spectral clustering) are susceptible to the effect of natural randomness and detected hundreds to tens of thousands of spurious patterns in datasets following H_0; many output spurious patterns in all of these datasets (i.e., $\kappa_{cluster} = 100$). Although DB_B and HDBSCAN with higher min_{size} values have relatively fewer number of spurious detections, we will soon show that the detection power of those versions with more strict criteria suffer greatly when true clusters do exist. Also, for deep embedding clustering, some clusters occasionally do not receive points in the final assignments after convergence (unlike spectral clustering, which often runs k-means as the final step, this method runs k-means during initialization). Finally, both Significant DBSCAN and Significant DBSCAN+ are able to robustly control the rate of spurious results, i.e., seven spurious patterns in total across 600 datasets for Significant DBSCAN and five spurious patterns for Significant DBSCAN+.

Improvements on solution quality for data following H_1: We varied both data size N and effect sizes es during data generation for each base landscape shown in Figure 8, i.e., "Shape," "Test," and "Dense." Tables 3 and 4 show the F1 scores for the three base landscapes with varying N and es, and Tables 5 and 6 show the ACC and NMI scores. The value in each cell is averaged over results from 100 datasets of the corresponding combination of parameter setting and base landscape. The highest score in each row is highlighted in bold. For F1 scores, a general trend we can observe is that Significant DBSCAN+ achieved the best solution quality in most of the scenarios (27 of 36 rows overall; 32 of 36 if we include Significant DBSCAN), including many big margins such as 10–20% increases in absolute F1 scores. Similar to k-means and expectation-maximization results from our preliminary work, spectral clustering and deep embedding clustering do not perform well here in the experiment as they are more of the partitioning-type of approaches, and their F1 scores drop quickly as the proportion of noise increases (e.g., results for "Shape" vs. "Test" and "Dense" base landscapes). Note that one advantage of deep embedding clustering is that it can more easily handle high-dimensional data such as images, which is outside the scope for this article. In the future we will explore an integration of deep embedding and Significant DBSCAN+ (Section 6). In addition, although the strict criteria in several versions of DBSCAN and HDBSCAN (i.e., DB_B and HDB_{200}) helped them to suppress the number of spurious results (Table 2), those criteria also cause them to miss lots of true clusters, leading to low F1 scores as shown in Tables 3 and 4. In contrast, Significant DBSCAN+ is able to simultaneously control the number of spurious detections and greatly improve the F1 scores. For ACC and NMI, we can observe similar trends where SigDB+ achieved the best ACC performance in 32 of 36 settings, and the best NMI performance in 31 of 36 settings (we selected the best-performing method in the five groups {DB_A, DB_B}, {Spectral}, {Deep}, {HDB_5, HDB_{100}, HDB_{200}}, and {SigDB, SigDB+} to show in Tables 5 and 6 for these two additional comparisons). As mentioned earlier in Section 5.1.3, "F1 score vs. ACC (or NMI)" is analogous to

Table 3. F1 Scores of Candidate Methods on Data with Varying N

	N	DB_A	DB_B	Spectral	Deep	HDB_5	HDB_{100}	HDB_{200}	SigDB	SigDB+
Shape	2,000	0.093	0.001	0.003	0.043	0.244	0.272	0.010	**0.520**	0.272
	4,000	0.251	0.005	0.072	0.025	0.157	0.700	0.242	0.457	**0.745**
	6,000	0.405	0.009	0.163	0.015	0.097	0.850	0.567	0.568	**0.886**
	8,000	0.409	0.012	0.065	0.040	0.077	**0.842**	0.745	0.663	0.796
	10,000	0.509	0.026	0.072	0.025	0.055	0.818	0.818	0.743	**0.989**
	16,000	0.568	0.073	0.089	0.079	0.027	0.842	0.853	0.909	**0.992**
Test	2,000	0.106	0.000	0.000	0.000	0.042	0.006	0.000	0.232	**0.240**
	4,000	0.211	0.006	0.000	0.000	0.043	**0.530**	0.003	0.289	0.482
	6,000	0.268	0.004	0.000	0.000	0.030	**0.588**	0.248	0.413	0.574
	8,000	0.275	0.010	0.000	0.000	0.024	0.394	0.621	0.480	**0.677**
	10,000	0.277	0.004	0.000	0.000	0.020	0.325	0.604	0.565	**0.784**
	16,000	0.296	0.004	0.000	0.000	0.011	0.195	0.367	0.766	**0.873**
Dense	2,000	0.273	0.054	0.008	0.000	0.187	0.000	0.000	0.261	**0.455**
	4,000	0.410	0.083	0.013	0.000	0.159	0.113	0.000	0.235	**0.681**
	6,000	0.443	0.149	0.009	0.001	0.127	0.381	0.001	0.434	**0.731**
	8,000	0.448	0.182	0.001	0.000	0.096	0.517	0.112	0.537	**0.814**
	10,000	0.497	0.159	0.001	0.000	0.081	0.577	0.251	0.640	**0.865**
	16,000	0.535	0.179	0.000	0.000	0.047	0.689	0.546	0.852	**0.898**

Table 4. F1 Scores of Candidate Methods on Data with Varying *es*

	es	DB_A	DB_B	Spectral	Deep	HDB_5	HDB_{100}	HDB_{200}	SigDB	SigDB+
Shape	(2,4)	0.301	0.023	0.000	0.003	0.006	0.599	0.373	0.408	**0.631**
	(3,6)	0.414	0.055	0.009	0.010	0.029	0.797	0.677	0.742	**0.978**
	(4,6)	0.523	0.022	0.062	0.014	0.053	0.872	0.820	0.746	**0.986**
	(6,8)	0.462	0.016	0.174	0.024	0.112	0.901	0.963	0.824	**0.994**
	(3,10)	0.420	0.059	0.118	0.069	0.043	0.788	0.608	**0.859**	0.804
	(4,8)	0.431	0.031	0.083	0.034	0.065	0.936	0.799	0.805	**0.987**
Test	(2,4)	0.024	0.009	0.000	0.000	0.004	0.009	0.114	0.058	**0.519**
	(3,6)	0.285	0.002	0.000	0.000	0.013	0.192	0.416	0.392	**0.521**
	(4,6)	0.283	0.004	0.000	0.000	0.019	0.296	0.647	0.534	**0.771**
	(6,8)	0.279	0.002	0.000	0.000	0.039	0.593	0.817	0.892	**0.958**
	(3,10)	0.280	0.006	0.003	0.000	0.018	0.372	0.401	**0.614**	0.516
	(4,8)	0.283	0.011	0.000	0.000	0.023	0.411	0.563	0.649	**0.773**
Dense	(2,4)	0.225	0.177	0.000	0.000	0.024	0.339	0.095	0.593	**0.608**
	(3,6)	0.426	0.179	0.002	0.000	0.057	0.467	0.281	**0.754**	0.628
	(4,6)	0.501	0.154	0.003	0.000	0.081	0.580	0.309	0.715	**0.840**
	(6,8)	0.536	0.153	0.038	0.000	0.155	0.655	0.404	0.534	**0.934**
	(3,10)	0.302	0.187	0.134	0.003	0.073	0.443	0.350	**0.708**	0.679
	(4,8)	0.446	0.185	0.040	0.000	0.093	0.589	0.348	**0.940**	0.870

"object vs. pixel" level measures in object detection, so different score distributions are expected. For example, in a dataset with four clusters of similar sizes, an incorrect clustering that puts all points into a single cluster might still get an ACC of 0.25 due to the overlap, but will get a zero F1 score. Overall, similar rankings are maintained among the measures.

Table 5. ACC and NMI of Candidate Methods on Data with Varying N

		Unsupervised clustering accuracy					Normalized mutual information				
	N	DB_A	Spectral	Deep	HDB_{100}	SigDB+	DB_A	Spectral	Deep	HDB_{100}	SigDB+
Shape	2,000	0.702	0.373	0.530	0.727	**0.723**	0.690	0.263	0.419	0.592	**0.713**
	4,000	0.824	0.606	0.535	0.850	**0.892**	0.762	0.574	0.425	0.748	**0.818**
	6,000	0.865	0.719	0.529	0.882	**0.920**	0.800	0.668	0.416	0.790	**0.839**
	8,000	0.867	0.617	0.537	**0.878**	0.814	**0.801**	0.622	0.435	0.793	0.698
	10,000	0.891	0.536	0.526	0.884	**0.945**	0.820	0.595	0.418	0.801	**0.867**
	16,000	0.899	0.450	0.479	0.898	**0.946**	0.830	0.546	0.397	0.813	**0.871**
Test	2,000	0.793	0.457	0.446	0.758	**0.824**	0.675	0.208	0.277	0.455	**0.682**
	4,000	0.835	0.443	0.447	**0.898**	0.890	0.704	0.406	0.287	0.708	**0.750**
	6,000	0.853	0.493	0.438	0.891	**0.905**	0.725	0.471	0.279	0.734	**0.768**
	8,000	0.853	0.490	0.441	0.855	**0.913**	0.723	0.483	0.285	0.712	**0.775**
	10,000	0.856	0.463	0.452	0.841	**0.924**	0.722	0.471	0.258	0.691	**0.793**
	16,000	0.861	0.436	0.483	0.828	**0.929**	0.729	0.462	0.236	0.677	**0.800**
Dense	2,000	0.783	0.377	0.366	0.491	**0.799**	**0.752**	0.331	0.385	0.435	0.742
	4,000	0.845	0.469	0.404	0.612	**0.868**	0.791	0.537	0.434	0.600	**0.798**
	6,000	0.856	0.559	0.407	0.741	**0.867**	**0.799**	0.603	0.424	0.718	0.797
	8,000	0.855	0.590	0.416	0.773	**0.893**	0.799	0.617	0.419	0.749	**0.821**
	10,000	0.869	0.610	0.392	0.791	**0.906**	0.808	0.621	0.374	0.759	**0.832**
	16,000	0.875	0.553	0.375	0.832	**0.907**	0.811	0.591	0.342	0.786	**0.834**

Table 6. ACC and NMI of Candidate Methods on Data with Varying es

		Unsupervised clustering accuracy					Normalized mutual information				
	es	DB_A	Spectral	Deep	HDB_{100}	SigDB+	DB_A	Spectral	Deep	HDB_{100}	SigDB+
Shape	(2,4)	**0.815**	0.499	0.369	0.765	0.791	**0.712**	0.513	0.232	0.692	0.682
	(3,6)	0.858	0.582	0.334	0.796	**0.877**	0.786	0.594	0.263	0.755	**0.802**
	(4,6)	0.870	0.606	0.325	0.798	**0.903**	0.809	0.618	0.257	0.768	**0.830**
	(6,8)	0.876	0.661	0.300	0.810	**0.925**	0.833	0.679	0.294	0.795	**0.867**
	(3,10)	0.845	0.648	0.323	0.832	**0.900**	0.794	0.660	0.268	0.796	**0.844**
	(4,8)	0.864	0.637	0.312	0.824	**0.918**	0.811	0.652	0.267	0.793	**0.853**
Test	(2,4)	**0.844**	0.419	0.432	0.844	0.835	0.735	0.488	0.263	0.726	**0.737**
	(3,6)	0.875	0.493	0.426	0.884	**0.934**	0.793	0.567	0.271	0.787	**0.846**
	(4,6)	0.889	0.540	0.407	0.894	**0.943**	0.820	0.596	0.270	0.812	**0.865**
	(6,8)	0.879	0.658	0.426	0.898	**0.959**	0.830	0.673	0.309	0.831	**0.896**
	(3,10)	0.900	0.609	0.527	0.900	**0.902**	0.820	0.633	0.355	0.800	**0.840**
	(4,8)	0.885	0.595	0.455	0.906	**0.951**	0.816	0.630	0.308	0.819	**0.878**
Dense	(2,4)	0.864	0.374	0.468	0.822	**0.866**	**0.655**	0.374	0.157	0.596	0.628
	(3,6)	0.867	0.447	0.453	0.849	**0.901**	0.714	0.453	0.167	0.685	**0.752**
	(4,6)	0.856	0.470	0.460	0.842	**0.923**	0.724	0.475	0.190	0.693	**0.793**
	(6,8)	0.837	0.535	0.417	0.868	**0.937**	0.742	0.534	0.205	0.751	**0.835**
	(3,10)	0.877	0.482	0.468	0.863	**0.918**	0.746	0.502	0.183	0.718	**0.795**
	(4,8)	0.859	0.488	0.439	0.858	**0.929**	0.734	0.497	0.184	0.718	**0.812**

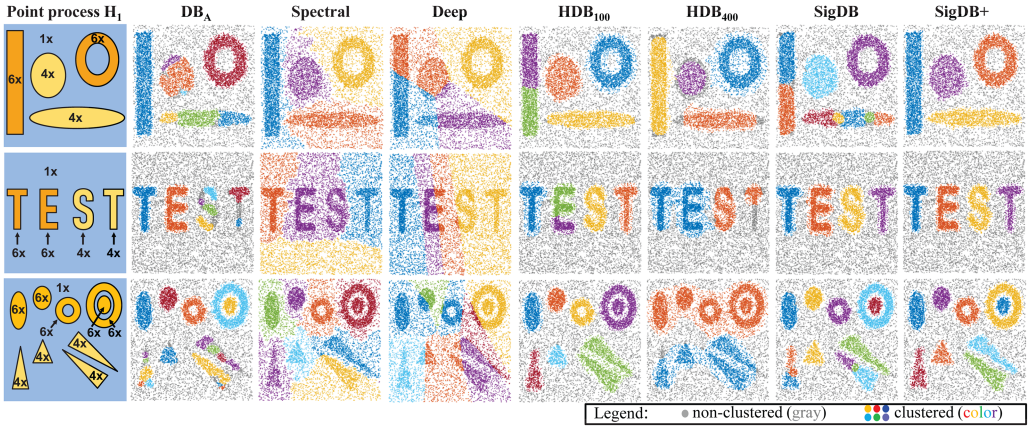


Fig. 9. Visualization of results on three sample realizations of point processes (data size $N = 10{,}000$). Results in Tables 3 and 6 are summarized from 3,600 realizations under 36 different combinations of N and es.

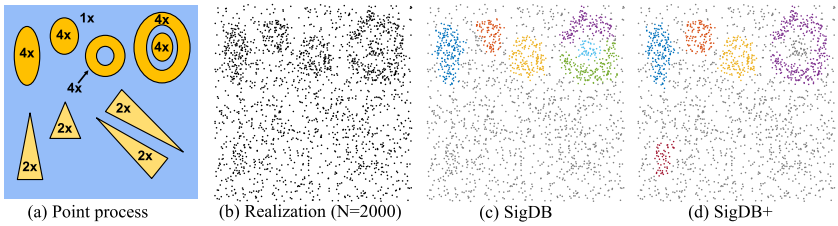


Fig. 10. Challenging scenarios with diminishing clustering signals. With a combination of low effect sizes and small number of points, it becomes difficult to confirm the significance, or to observe, true clusters by both Significant DBSCAN and Significant DBSCAN+.

Significant DBSCAN vs. Significant DBSCAN+: Compared to our preliminary work—Significant DBSCAN [42]—we can see that the new extensions in Significant DBSCAN+ can improve the solution quality (Tables 3 and 4) especially in scenarios where the effect sizes of clusters in es are relatively close. As illustrated in Section 3.3, Significant DBSCAN has difficulty in separating out clusters with moderate overlaps in their density distributions and generates shattered clusters in outputs. However, it performs fine for settings where the effect sizes of clusters are very different (e.g., $es = (3, 10)$ in Table 4). Significant DBSCAN+ maintains good performances overall by addressing this limitation.

Qualitative evaluation: Figure 9 shows the visual comparisons of the results generated by the candidate methods on one example realization of the point process (statistics in Tables 3 and 4 are based on 3,600 realizations under 36 different scenarios). Particularly, we can see that the cluster shattering issues in Significant DBSCAN caused by overlaps in cluster density distributions are reduced with the proposed extensions in Significant DBSCAN+. Spectral clustering is given the correct number of clusters as input, which is often unknown in practice (for the third row, the ninth cluster returned by spectral clustering only has two points and is not very visible).

Finally, Figure 10 shows a challenging example where it is difficult for both Significant DBSCAN and Significant DBSCAN+ to detect the true clusters. This happens when both the effect size and the number of points are small (i.e., weak signals of clusters), making it statistically difficult to

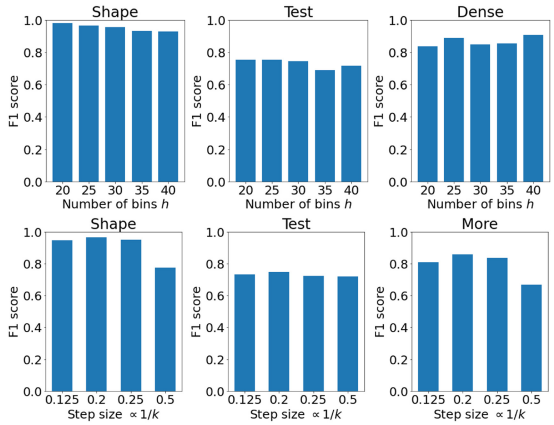


Fig. 11. Sensitivity analysis on the number of scales k (represented by step-size $3/k$ between ϵ_{max} and ϵ_{min}) for the multi-scale DBSCAN sub-routine, and the number of bins h for OaaT density selection.

observe the signals and confirm their significance. In other words, if the cumulative probability of a cluster is largely dominated by that of the background noise in the point process, it will require a large number of observations to confirm its significance.

5.1.5 Sensitivity Analysis. Here we further evaluate the sensitivity of the proposed Significant DBSCAN+ using F1 scores. First, according to the results in Tables 3 and 4, the solution quality in general tends to increase as the data size N increases. This trend is intuitive as a greater number of observations makes it statistically easier to capture the true clusters and confirm their significance. Similarly, a sharper contrast between the probability densities inside and outside clusters also makes them easier to detect. Comparing results on the three different base landscapes, the method tends to perform better on data with clusters having relatively greater volumes (compared to the background). This can be explained similarly from the statistical perspective, since a greater volume naturally increases the number of observations (both absolutely and relatively) in the clustered area, making the clusters easier to observe and pass the test [39, 43].

In addition, Figure 11 shows the F1 scores of Significant DBSCAN with varying hyperparameters, i.e., the number of bins h in the histogram for OaaT-based density selection (Section 4.2), and the number of scales k in multi-scale DBSCAN (Section 4.3). Each F1 score is computed over 100 datasets generated using the default values of $N = 10{,}000$ and $es = (4, 6)$ for each of the three base landscapes (Figure 8). As we can see in Figure 11 (top), the proposed approach has a stable performance over different number of bins in the experiment. In general, the OaaT approach does prefer the cluster size to be bigger than $1/h$ proportion of the entire dataset (e.g., 0.025 for $h = 40$) so that it is easier to observe the local peak. Thus, the choice of h may be adjusted for scenarios where such tiny clusters exist. Figure 11 (bottom) shows the F1 scores of the approach with different number of scales k, represented by step size $\propto 1/k$. The trend is that when k is relatively large (i.e., small step size), the performance of Significant DBSCAN remains stable, whereas if k is very small, its F1 score decreases. The main reason is that as k reduces to 1, multi-scale DBSCAN reduces to DBSCAN, which takes away the associated improvements.

5.2 Computational Performance

The computational enhancements in this extension is a generalization of the dual-convergence algorithm in our preliminary work on Significant DBSCAN [42] (summary in Section 3.1.4). It

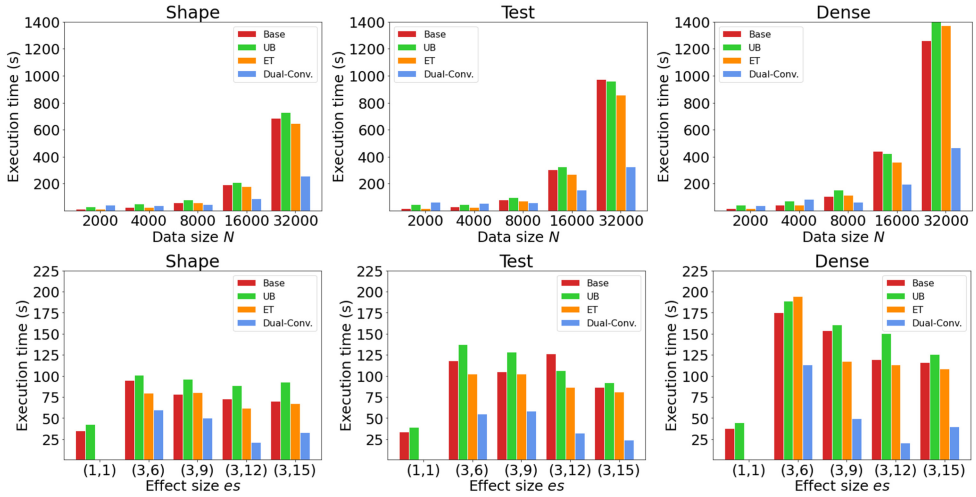


Fig. 12. Execution time on varying data sizes N and effect sizes es for the baseline (Base), upper-bound-only (UB), early-termination-only (ET) and the full dual-convergence algorithm.

extends the single-scale based local search and upper-bound computation to a multi-scale version. The controlled data are generated using the same set of landscapes shown in Figure 8. Since the trends remain consistent with our preliminary work, here we will just briefly summarize the new results to avoid redundancy.

Figure 12 shows the execution time for four candidate approaches, i.e., baseline, baseline + bound-pruning, baseline + early-termination, and the dual-convergence algorithm (baseline + dynamic coordination between bound-pruning and early-termination). The time is evaluated on datasets with different data sizes N and effect sizes es. The default number of points and effect sizes (when not being varied) are set to $N = 10{,}000$ and $es = (3, 6)$.

As we can see, the dual-convergence algorithm is able to consistently reduce the computational cost, whereas the bound-pruning based approach or the early-termination approach alone cannot effectively speed-up the detection in general. Especially, without the coordination provided by the dual-convergence process, the bound-based pruning itself may result in additional cost due to the overhead in bound calculation. As shown by the first groups in Figure 12 (bottom), the early-termination algorithm itself does perform very well when effect sizes are set to $es = (1, 1)$. This is because when effect size equals 1, the clustered point process reduces to the homogeneous point process H_0. Since early-termination is very efficient when no significant cluster exists, it is able to greatly reduce the cost for this special case. However, such time reduction quickly fades away as true clusters start to exist (i.e., effect size > 1). In contrast, the dual-convergence algorithm maintains the speed-up across various effect sizes in the experiments.

6 CONCLUSIONS AND FUTURE WORK

We proposed a Significant DBSCAN+ to extend our preliminary work [42] on the detection of statistically significant clusters that are density-connected. Specifically, the extension strengthened our previous work's capacity to find significant clusters with varying densities via a newly developed OaaT density selection, multi-scale DBSCAN sub-routine and a VISIT algorithm for selecting a stable set of candidates. Computation-wise, we also generalized the dual-convergence algorithm in our previous work for the proposed extensions. Through controlled experiments, we showed

that the incorporation of statistical rigor is a powerful mechanism, and Significant DBSCAN+ is able to robustly eliminate spurious patterns and greatly improve the solution quality. In addition, this extension also outperformed our previous work especially in scenarios where the density distributions of clusters overlap. The generalized dual-convergence algorithm also greatly reduced the computational cost.

In future work, we plan to first explore improvements of Significant DBSCAN+ for higher-dimensional datasets, where both new definitions of null hypotheses (e.g., H_0 may not be homogeneous along all dimensions) and corresponding computational structures will be investigated. Specifically, we will explore ways to leverage embedding methods from deep clustering approaches to handle features in high-dimensions. In addition, we will further explore statistically robust formulations and computational techniques for other clustering or data-partitioning sub-routines such as HDBSCAN, Chameleon, and spectral clustering, which require new modeling approaches as clusters with different densities are returned at the same time. Finally, we will explore spatial big data extensions in distributed environments (e.g., Apache Sedona).

REFERENCES

[1] 2020. HDBSCAN. Retrieved from https://hdbscan.readthedocs.io/en/latest/index.html.

[2] 2020. National Cancer Institute. Retrieved from https://surveillance.cancer.gov/satscan/.

[3] 2020. SaTScan. Retrieved from https://www.satscan.org/.

[4] 2021. Keras implementation for Deep Embedding Clustering (DEC). Retrieved from https://github.com/XifengGuo/DEC-keras.

[5] David Arthur and Sergei Vassilvitskii. 2006. *k-means++: The Advantages of Careful Seeding*. Technical Report. Stanford.

[6] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. 2018. Spatio-temporal data mining: A survey of problems and methods. *ACM Comput. Surv.* 51, 4 (2018), 1–41.

[7] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 160–172.

[8] Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. 2015. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data* 10, 1 (2015), 1–51.

[9] Marcelo Azevedo Costa, Renato Martins Assunçao, and Martin Kulldorff. 2012. Constrained spanning tree algorithms for irregularly-shaped spatial clustering. *Comput. Stat. Data Anal.* 56, 6 (2012), 1771–1783.

[10] Luiz Duczmal, Anderson Ribeiro Duarte, and Ricardo Tavares. 2009. Extensions of the scan statistic for the detection and inference of spatial clusters. In *Scan Statistics*. Springer, 153–177.

[11] Luiz Duczmal, Martin Kulldorff, and Lan Huang. 2006. Evaluation of spatial scan statistics for irregularly shaped clusters. *J. Comput. Graph. Stat.* 15, 2 (2006), 428–442.

[12] Emre Eftelioglu et al. 2014. Ring-shaped hotspot detection: A summary of results. In *Proceedings of the 2014 IEEE International Conference on Data Mining (ICDM'14)*. IEEE, 815–820.

[13] Junhao Gan and Yufei Tao. 2015. DBSCAN revisited: mis-claim, un-fixability, and approximation. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 519–530.

[14] Joseph Glaz and Markos V. Koutras. 2019. *Handbook of Scan Statistics*. Springer.

[15] Zhenyu Huang, Peng Hu, Joey Tianyi Zhou, Jiancheng Lv, and Xi Peng. 2020. Partially view-aligned clustering. *Advances in Neural Information Processing Systems* 33 (2020), 2892–2902. https://proceedings.neurips.cc/paper/2020/hash/1e591403ff232de0f0f139ac51d99295-Abstract.html.

[16] George Karypis, Eui-Hong Han, and Vipin Kumar. 1999. Chameleon: Hierarchical clustering using dynamic modeling. *Computer* 32, 8 (1999), 68–75.

[17] Martin Kulldorff. 1997. A spatial scan statistic. *Comm. Stati. Theory Methods* 26, 6 (1997), 1481–1496.

[18] Xiao-Zhou Li, Jin-Feng Wang, Wei-Zhong Yang, Zhong-Jie Li, and Sheng-Jie Lai. 2011. A spatial scan statistic for multiple clusters. *Math. Biosci.* 233, 2 (2011), 135–142.

[19] Jacklin F. Mosha, Hugh J. W. Sturrock, Brian Greenwood, Colin J. Sutherland, Nahla B. Gadalla, Sharan Atwal, Simon Hemelaar, Joelle M. Brown, Chris Drakeley, Gibson Kibiki, et al. 2014. Hot spot or not: A comparison of spatial statistical methods to predict prospective malaria infections. *Malaria J.* 13, 1 (2014), 53.

[20] Daniel B. Neill. 2006. Detection of spatial and spatio-temporal clusters. Ph.D. Dissertation. Carnegie Mellon University.

[21] Daniel B. Neill. 2018. *Bayesian Scan Statistics*. Springer, New York, NY, 1–21.

[22] Daniel B. Neill and Andrew W. Moore. 2004. Rapid detection of significant spatial clusters. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 256–265.

[23] Antonio Cavalcante Araujo Neto, Joerg Sander, Ricardo J. G. B. Campello, and Mario A. Nascimento. 2017. Efficient computation of multiple density-based clustering hierarchies. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'17)*. IEEE, 991–996.
[24] Andrew Y. Ng, Michael I. Jordan, Yair Weiss, et al. 2002. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* 2 (2002), 849–856.
[25] Feiping Nie, Xiaoqian Wang, and Heng Huang. 2014. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 977–986.
[26] Ganapati P. Patil and Charles Taillie. 2004. Upper level set scan statistic for detecting arbitrarily shaped hotspots. *Environ. Ecol. Stat.* 11, 2 (2004), 183–197.
[27] Chong Peng, Zhao Kang, Shuting Cai, and Qiang Cheng. 2018. Integrate and conquer: Double-sided two-dimensional k-means via integrating of projection and manifold construction. *ACM Trans. Intell. Syst. Technol.* 9, 5 (2018), 1–25.
[28] Chong Peng, Qian Zhang, Zhao Kang, Chenglizhao Chen, and Qiang Cheng. 2021. Kernel two-dimensional ridge regression for subspace clustering. *Pattern Recogn.* 113 (2021), 107749.
[29] Chong Peng, Zhilu Zhang, Zhao Kang, Chenglizhao Chen, and Qiang Cheng. 2021. Nonnegative matrix factorization with local similarity learning. *Inf. Sci.* 562 (2021), 325–346.
[30] Xi Peng, Jiashi Feng, Joey Tianyi Zhou, Yingjie Lei, and Shuicheng Yan. 2020. Deep subspace clustering. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 12 (2020), 5509–5521.
[31] Mingjie Qian and Chengxiang Zhai. 2013. Robust unsupervised feature selection. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. Citeseer, 1621–1627. https://dl.acm.org/doi/10.5555/2540128.2540361.
[32] Douglas A. Reynolds. 2009. Gaussian mixture models. In *Encyclopedia of Biometrics*, Stan Z. Li and Anil Jain (Eds.). Springer US, 659–663. DOI: 10.1007/978-0-387-73003-5_196
[33] Camilo Rivera and Guenther Walther. 2013. Optimal detection of a jump in the intensity of a Poisson process or in a density with likelihood ratio statistics. *Scand. J. Stat.* 40, 4 (2013), 752–769.
[34] Kaspar Rufibach and Guenther Walther. 2010. The block criterion for multiscale inference about a density, with applications to other multiscale problems. *J. Comput. Graph. Stat.* 19, 1 (2010), 175–190.
[35] Shashi Shekhar, Steven K. Feiner, and Walid G. Aref. 2015. Spatial computing. *Commun. ACM* 59, 1 (2015), 72–81.
[36] Shashi Shekhar, Zhe Jiang, Reem Y. Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. 2015. Spatiotemporal data mining: A computational perspective. *ISPRS Int. J. Geo-Inf.* 4, 4 (2015), 2306–2338.
[37] Xun Tang et al. 2017. Significant linear hotspot discovery. *IEEE Trans. Big Data* 3, 2 (2017), 140–153.
[38] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Stat. Comput.* 17, 4 (2007), 395–416.
[39] Guenther Walther et al. 2010. Optimal and fast detection of spatial clusters with scan statistics. *Ann. Stat.* 38, 2 (2010), 1010–1033.
[40] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on Machine Learning*. 478–487.
[41] Yiqun Xie and Shashi Shekhar. 2019. A nondeterministic normalization based scan statistic (NN-scan) towards robust hotspot detection: A summary of results. In *Proceedings of the SIAM International Conference on Data Mining (SDM'19)*.
[42] 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.
[43] Yiqun Xie and Shashi Shekhar. 2020. A unified framework for robust and efficient hotspot detection in smart cities. *ACM Trans. Data Sci.* 1, 3 (2020), 1–29.
[44] Yiqun Xie, Shashi Shekhar, and Yan Li. 2021. Statistically-robust clustering techniques for mapping spatial hotspots: A survey. arXiv:2103.12019. Retrieved from https://arxiv.org/abs/2103.12019.
[45] Dongkuan Xu and Yingjie Tian. 2015. A comprehensive survey of clustering algorithms. *Ann. Data Sci.* 2, 2 (2015), 165–193.
[46] Rui Xu and Donald Wunsch. 2005. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* 16, 3 (2005), 645–678.
[47] Qian Zhang and Chong Peng. 2020. Feature selection embedded robust k-means. *IEEE Access* 8 (2020), 166164–166175. https://ieeexplore.ieee.org/abstract/document/9187813.

Received March 2021; revised July 2021; accepted July 2021