DNA: A Distribution-and-Aggregation Solution for Spatiotemporal K-function-based Analysis

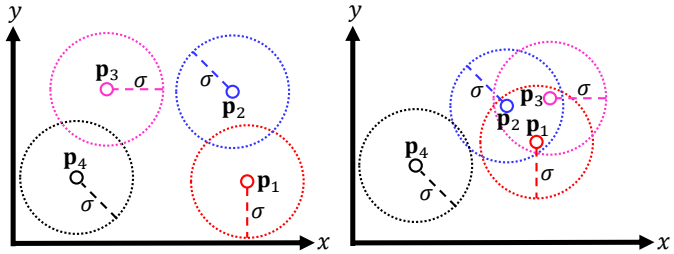
Tsz Nam Chan*, Bojian Zhu†, Dingming Wu*, Renchi Yang†, Ruisheng Wang‡
*College of Computer Science and Software Engineering, Shenzhen University
{edisonchan, dingming}@szu.edu.cn
†Department of Computer Science, Hong Kong Baptist University
csbjzhu@comp.hkbu.edu.hk, renchi@hkbu.edu.hk
‡School of Architecture and Urban Planning, Shenzhen University
ruiswang@szu.edu.cn

***Abstract*—Generating a spatiotemporal K-function plot is frequently adopted to analyze point patterns by domain experts in a wide range of application domains, including criminology, transportation science, urban planning, and epidemiology. However, with the high worst-case time complexity of computing a spatiotemporal K-function plot, the state-of-the-art methods are unable to efficiently (or feasibly) support this tool. To overcome this issue, we develop Distribution-aNd-Aggregation (DNA), which is the first solution that can reduce the worst-case time complexity for supporting this tool. Experiment results on four large-scale location datasets show that DNA achieves speedups of 4.58x to 57.42x over the state-of-the-art methods, without incurring significant space overhead. The implementation of all methods can be found in https://github.com/edisonchan2013928/DNA.**

I. Introduction

Point pattern analysis [1], [2] is an important field for many application domains, including geography, transportation science, epidemiology, criminology, urban planning, cheminformatics, and bioinformatics. Among most of the statistical tools in this field, spatial K-function (or Ripley's K-function) [1], [3] is the fundamental tool to analyze the correlation between data points in a location dataset P, which counts all data points that are within the spatial threshold σ from each data point in P. Figure 1 depicts the concept of spatial K-function. In Figure 1a, since there is no data point within the spatial threshold σ from each data point, the spatial K-function value of this dataset is 0. Observe from Figure 1b that the sets $\{\mathbf{p}_2, \mathbf{p}_3\}$, $\{\mathbf{p}_1, \mathbf{p}_3\}$, and $\{\mathbf{p}_1, \mathbf{p}_2\}$ are within the spatial threshold σ from $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, respectively, and no data point is within the spatial threshold σ from $\mathbf{p}_4$. As such, the spatial K-function value for this dataset is 6. Note that the dataset tends to be clustered (or dispersed) if the spatial K-function value is larger (or smaller).

Spatial K-function has been extensively used by domain experts, including criminologists [4], [5], [6], [7], [8], transportation experts [7], [9], [10], urban planners [11], [12], [13], [14], epidemiologists [15], [16], [17], [18], and molecular scientists [19], [20], [21], [22], [23], [24], [25], [26], [27], to analyze their location datasets and perform meta analysis. To adopt this tool, domain experts need to generate a spatial K-function plot (see Figure 2) by (1) choosing S spatial thresholds, i.e., σ_1, σ_2,..., σ_S, and randomly generating L datasets with the same size as the original location dataset P, (2) computing the spatial K-function values for $L+1$ datasets (including the original one) with each spatial threshold σ (where σ can be σ_1, σ_2,..., σ_S), (3) plotting the black curve (spatial K-function values of the original dataset with respect to different spatial thresholds σ), and (4) plotting the red dotted curve and the blue dotted curve, which, for each spatial threshold σ, represent the minimum and maximum spatial K-function values, respectively, of L random datasets. Once the black curve is above the blue dotted curve, they conclude that the location dataset contains meaningful clusters/hotspots



(a) Dispersed (b) Clustered

Fig. 1: Illustration of spatial K-function.

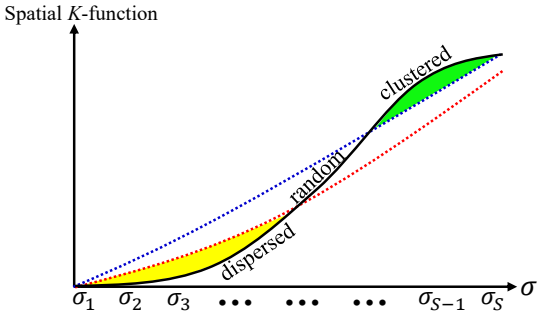
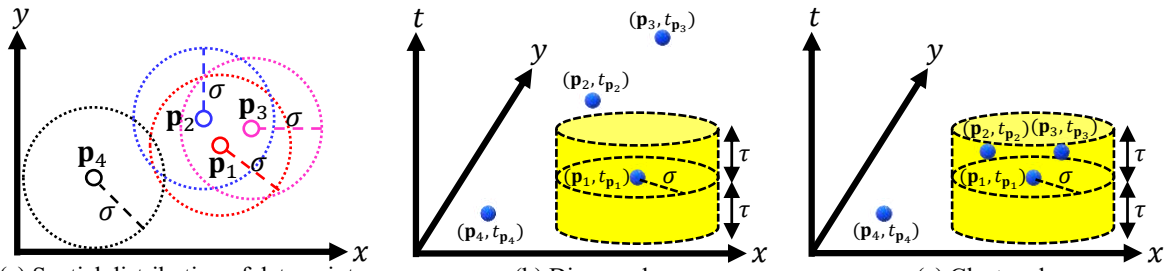


Fig. 2: A spatial K-function plot.

This work was supported by the Natural Science Foundation of China under grants 23IAA00610, 62572326, 62372308, and 62302414, Scientific Foundation for Youth Scholars of Shenzhen University, the Guangdong and Hong Kong Universities "1+1+1" Joint Research Collaboration Scheme, project No.: 2025A0505000002, and the Key Technological Innovation Program of Ningbo City under Grant No. 2024Z297. Dingming Wu is the corresponding author of this paper.



(a) Spatial distribution of data points (b) Dispersed (c) Clustered

Fig. 3: Spatial K-function, which ignores the timestamp of each data point, can possibly provide misleading cluster properties (i.e., dispersed or clustered) for a location dataset with the same spatial distribution (in (a)) but two temporal distributions (in (b) and (c)).

with respect to those spatial thresholds (see the green region of Figure 2). In contrast, they regard the dataset to be either random (see Figure 2) or dispersed (see the yellow region of Figure 2) for those spatial thresholds. Due to the popularity of spatial K-function, many geographical and statistical software suites, including ArcGIS [28], CrimeStat [29], spatstat [30], and PySAL [31], can also support this tool.

However, since many geographical phenomena (e.g., different waves of disease outbreak [32], [33] and repeated patterns of crime events [34], [35], [36]) depend on time, one major drawback for using spatial K-function is that this tool does not consider any temporal information (i.e., the timestamp) of each location data point. Hence, this tool may result in misleading interpretation for geographical applications. Consider Figures 3b and c as an example of two disease outbreak location datasets with the same spatial distribution (see Figure 3a) and two temporal distributions. Although the data points $(\mathbf{p}_1, t_{\mathbf{p}_1})$, $(\mathbf{p}_2, t_{\mathbf{p}_2})$, and $(\mathbf{p}_3, t_{\mathbf{p}_3})$ in Figure 3b are close to each other in terms of spatial coordinates (see Figure 3a), these data points are far away from each other in terms of timestamps, which may indicate the sporadic cases. In contrast, the data points $(\mathbf{p}_1, t_{\mathbf{p}_1})$, $(\mathbf{p}_2, t_{\mathbf{p}_2})$, and $(\mathbf{p}_3, t_{\mathbf{p}_3})$ in Figure 3c are spatiotemporally close to each other, which indicates that there is a valid disease outbreak cluster/hotspot for this dataset. As such, spatial K-function is unable to distinguish these two cases, possibly leading to incorrect conclusion for the cluster property of a dataset (e.g., regard the dataset in Figure 3b, based on Figure 3a, as clustered).

To overcome this weakness, many domain experts [35], [36], [37] have pointed out that incorporating time information into the K-function-based point pattern analysis can provide more meaningful and accurate results. Hence, they have adopted the advanced point pattern analysis tool, called spatiotemporal K-function [38], [39], [32], [33], [34], [37], [35], [36], which aims to count all data points that are simultaneously within the spatial threshold σ and the temporal threshold τ from each data point (see Figures 3b and c). Therefore, instead of plotting a two-dimensional spatial K-function plot (see Figure 2), domain experts obtain a three-dimensional spatiotemporal K-function plot (see Figure 4), by computing multiple spatiotemporal K-functions with S spatial thresholds, i.e., σ_1, σ_2,..., σ_S, and T temporal thresholds, i.e., τ_1, τ_2,..., τ_T, with the original dataset (see the green surface of Figure 4) and L random datasets (see the red and blue surfaces of Figure 4). Once the green surface is above the blue surface, the location dataset tends to have meaningful clusters/hotspots with respect to those spatial and temporal thresholds. In contrast, this dataset is either random or dispersed. As a remark, we will provide a detailed case study for this dataset (and the corresponding plot) in Section V-E.

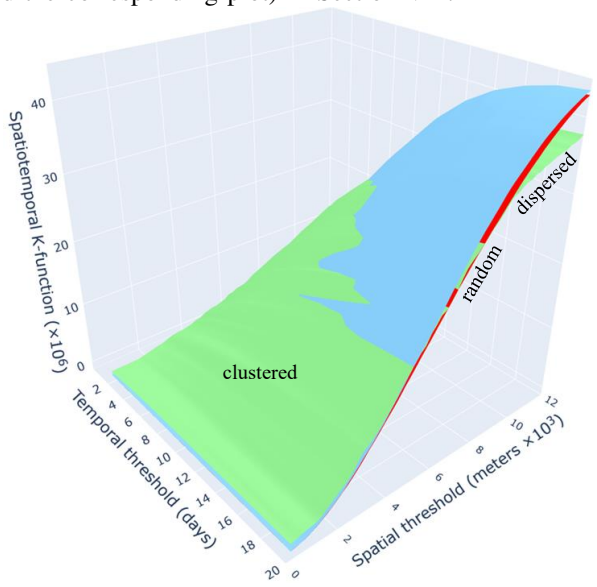


Fig. 4: A spatiotemporal K-function plot[1] for the COVID-19 cases in the north district of Hong Kong [40], where, for each pair of a spatial threshold and a temporal threshold (σ, τ), the red surface and the blue surface represent the minimum and the maximum spatiotemporal K-function values, respectively, of L random datasets and the green surface denotes the spatiotemporal K-function value of the original dataset for each (σ, τ)-pair.

However, generating a spatiotemporal K-function plot is time-consuming, which takes $O(LSTn^2)$ time. Therefore, this point pattern analysis tool cannot be scalable to support large-scale datasets (i.e., large n), a large number of spatial thresholds S, and a large number of temporal thresholds T, which has repeatedly been complained by various domain experts within the recent two decades [38], [44].

[1]In this example, we select $S = 60$, which are 200m, 400m,..., 12000m, $T = 20$, which are 1 day, 2 days,..., 20 days, and $L = 2$ for generating this plot.

TABLE I: Theoretical results for generating an exact spatiotemporal K-function plot, where n, L, S, and T denote the number of data points of a spatiotemporal dataset, the number of random datasets, the number of spatial thresholds, and the number of temporal thresholds, respectively.

Threshold interval type	Method	Time complexity	Space complexity	References
Fixed threshold intervals	RQS	$O(LSTn^2)$	$O(nL+ST)$	Section III-B ([41], [42], [43])
	DNA	$O(Ln^2+LSTn)$ (Theorem 1)	$O(nL+ST)$ (Theorem 3)	Section IV-B and Section IV-D
Multiple threshold intervals	RQS	$O(LSTn^2)$	$O(nL+ST)$	Section III-B ([41], [42], [43])
	DNA	$O(Ln^2 \log ST+LSTn)$ (Theorem 2)	$O(nL+ST)$ (Theorem 3)	Section IV-C and Section IV-D

Therefore, we ask a question in this paper. *Can we reduce the worst-case time complexity of generating an exact spatiotemporal K-function plot, without increasing the space complexity?* To provide an affirmative answer for this question, we propose a pioneering solution, called Distribution-aNd-Aggregation (DNA), which successfully reduces the worst-case time complexity for generating an exact spatiotemporal K-function plot with the commonly used fixed threshold intervals (see Figure 4), i.e., $\sigma_2-\sigma_1=\sigma_3-\sigma_2=...=\sigma_S-\sigma_{S-1}$ and $\tau_2-\tau_1=\tau_3-\tau_2=...=\tau_T-\tau_{T-1}$, from $O(LSTn^2)$ to $O(Ln^2+LSTn)$. In addition, we also extend DNA to generate this plot with multiple threshold intervals (e.g., $\sigma_1=100$m, $\sigma_2=200$m, $\sigma_3=500$m, $\sigma_4=1000$m, $\sigma_5=2000$m) with a slight overhead, i.e., $O(Ln^2 \log ST+LSTn)$ time, which is still faster than $O(LSTn^2)$. Furthermore, DNA retains the same space complexity, which is $O(nL+ST)$, no matter which threshold interval type we consider. Table I shows the theoretical results of all exact methods, including the state-of-the-art range-query-based solution (RQS) (will be discussed in Section III-B) and DNA, for generating a spatiotemporal K-function plot. Experiment results on four large-scale spatiotemporal datasets show that DNA achieves speedups of 4.58x to 57.42x over the state-of-the-art methods with nearly no space overhead. Here, we outline three main contributions of this paper.

- We develop the first complexity-reduced solution, namely DNA, for generating a spatiotemporal K-function plot with fixed threshold intervals and multiple threshold intervals.
- We utilize four large-scale location datasets, which belong to different categories, to verify that DNA can significantly outperform the state-of-the-art methods in terms of practical efficiency.
- We conduct a case study for showing how domain experts simultaneously use the spatiotemporal K-function plot with another point pattern analysis tool, called spatiotemporal kernel density visualization (STKDV), to analyze the COVID-19 cases of the north district in Hong Kong.

The rest of the paper is structured as follows. We first review the related work in Section II. Then, we discuss our problem and the state-of-the-art solution in Section III. Next, we illustrate our solution, DNA, in Section IV. After that, we present the experiment results of all methods in Section V. Lastly, we conclude this paper in Section VI.

II. Related Work

We review three camps of research studies that are closely related to this work.

Efficient methods for supporting K-function-related problems. In the literature, many efficient methods [45], [46], [47], [48], [37], [49] have been developed for improving the efficiency of solving K-function-related problems. Despite this, most of these research studies either combine the parallel/distributed approach [46], [48], [37], [49] or the modern hardware approach (e.g., GPU [49]) with existing methods, which cannot reduce the time complexity for solving the K-function-related problems. Furthermore, these methods consume many computational resources (e.g., 384 CPUs and 96 GPUs have been adopted in [49]). However, domain experts (e.g., criminologists) normally adopt the off-the-shelf software packages, e.g., ArcGIS [28] and spatstat [30], for analyzing their location datasets, who may not have enough computational resources. In addition, utilizing the parallel/distributed and hardware-based approaches for improving the efficiency of generating a spatiotemporal K-function plot is also orthogonal to this work. Recently, Chan et al. [45] and Rakshit et al. [47] propose the complexity-reduced algorithms for computing the network K-function, which is the variant of our problem. However, all these methods mainly adopt the properties of road networks (e.g., sharing the shortest path distances [47] and maintaining multiple sets of data points on each road [45]) in order to reduce the time complexity of computing the network K-function. Therefore, these two research studies cannot be extended for generating a spatiotemporal K-function plot since all data points of our problem lie in a three-dimensional space (see Figures 3b and c). Furthermore, unlike our work, none of these two research studies [45], [47] considers the optimization opportunity with multiple spatial and temporal thresholds.

Efficient methods for supporting kernel density visualization and its variants. In recent decades, many research studies [50], [51], [52], [53], [54], [55], [56], [57], [58], [59] have been proposed for supporting another important point pattern analysis tool, called kernel density visualization (KDV), which is based on computing the kernel aggregation with respect to data points. However, most of these research studies either (1) cannot reduce the time complexity [51], [54], [58], [59] or (2) can only provide the approximate result [51], [52], [53], [54], [55], [56], [57], [58], [59] for solving the KDV problem. Note that the state-of-the-art work [50] adopts the properties of pixels (e.g., all pixels in a row have the same y-coordinate) to develop the complexity-reduced solution for exactly solving the KDV problem. However, since all data points for our problem do not have these properties, this solution [50] cannot be extended for solving our problem. Recently, Chan et al. [60], [61], [62] further develop complexity-reduced algorithms for

solving the KDV variants, which are network kernel density visualization [60], [61] and spatiotemporal kernel density visualization [62]. Like [50], both [60], [61] and [62] mainly adopt the properties of lixels (i.e., the pixel in a road network) and the properties of voxels (three-dimensional small cubes), respectively, to develop these algorithms, which cannot be extended for generating a spatiotemporal K-function plot (as the data points in our problem do not have these properties).

Efficient indexing methods. Recall from Figures 3b and c that we need to count those data points that are simultaneously within the spatial and temporal thresholds, σ and τ, respectively, from each data point in order to compute the spatiotemporal K-function. Therefore, many indexing structures, e.g., kd-tree [41], ball-tree [42], and range-tree [43], can be used to boost the efficiency for computing a spatiotemporal K-function. However, all of them cannot reduce the time complexity for generating a spatiotemporal K-function plot (see Figure 4), which will be discussed in Section III-B in detail. Moreover, unlike our work, all these methods do not exploit the optimization opportunity for multiple spatial and temporal thresholds. Therefore, they provide the inferior efficiency performance compared with our methods.

III. Preliminaries

We first provide the formal definition of generating a spatiotemporal K-function plot in Section III-A. Then, we illustrate the state-of-the-art method, called range-query-based solution (RQS), for solving this problem in Section III-B. Table II summarizes the commonly used symbols of this paper.

TABLE II: Commonly used symbols.

Symbol	Description
P	Spatiotemporal dataset
R_1, R_2,..., R_L	L random datasets
n	Number of data points
S/T	Number of spatial/temporal thresholds
$d(\mathbf{p}_i, \mathbf{p}_j)$/$d(t_{\mathbf{p}_i}, t_{\mathbf{p}_j})$	Euclidean distance
σ/σ_u	Spatial threshold ($1 \leq u \leq S$)
τ/τ_v	Temporal threshold ($1 \leq v \leq T$)
$K_P(\sigma, \tau)$	Spatiotemporal K-function (for P)
$\mathcal{L}(\sigma, \tau)$/$\mathcal{U}(\sigma, \tau)$	Minimum/Maximum Spatiotemporal K-function values for L random datasets
$R_{\sigma_u, \tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})$	Range query set of $(\mathbf{p}_i, t_{\mathbf{p}_i})$
$\mathbb{G}$	Grid (with size $S \times T$)
$\mathbb{PC}$	Prefix count array (with size $S \times T$)

A. Problem Definition

In order to compute a spatiotemporal K-function for a dataset, we need to count all data points that are within a spatial threshold σ and a temporal threshold τ (see the yellow cylinders in Figures 3b and c), which is stated in Definition 1.

Definition 1. *Given a spatiotemporal dataset $P = \{(\mathbf{p}_1, t_{\mathbf{p}_1}), (\mathbf{p}_2, t_{\mathbf{p}_2}), ..., (\mathbf{p}_n, t_{\mathbf{p}_n})\}$ with size n, a spatial threshold σ, and a temporal threshold τ, the spatiotemporal K-function $K_P(\sigma, \tau)$ is defined as follows.*

$$K_P(\sigma, \tau) = \sum_{(\mathbf{p}_i, t_{\mathbf{p}_i}) \in P} \sum_{\substack{(\mathbf{p}_j, t_{\mathbf{p}_j}) \in P \\ j \neq i}} \mathcal{I}(d(\mathbf{p}_i, \mathbf{p}_j) \leq \sigma, d(t_{\mathbf{p}_i}, t_{\mathbf{p}_j}) \leq \tau) \quad (1)$$

where $\mathcal{I}$ and d denote the indicator function and the Euclidean distance function, respectively.

With the definition of the spatiotemporal K-function, we formally define the problem of generating a spatiotemporal K-function plot (see Figure 4) in Problem 1.

Problem 1. *Given a location dataset P with size n, L random datasets R_1, R_2,..., R_L (with the same size n), S spatial thresholds σ_1, σ_2,..., σ_S ($\sigma_1 \leq \sigma_2 \leq ... \leq \sigma_S$), and T temporal thresholds τ_1, τ_2,..., τ_T ($\tau_1 \leq \tau_2 \leq ... \leq \tau_T$), generating a spatiotemporal K-function plot needs to compute $K_P(\sigma, \tau)$ (see Equation 1), $\mathcal{L}(\sigma, \tau)$ (see Equation 2), and $\mathcal{U}(\sigma, \tau)$ (see Equation 3) for each pair of (σ, τ), where σ can be σ_1, σ_2,..., σ_S and τ can be τ_1, τ_2,..., τ_T.*

$$\mathcal{L}(\sigma, \tau) = \min(K_{R_1}(\sigma, \tau), K_{R_2}(\sigma, \tau), ..., K_{R_L}(\sigma, \tau)) \quad (2)$$

$$\mathcal{U}(\sigma, \tau) = \max(K_{R_1}(\sigma, \tau), K_{R_2}(\sigma, \tau), ..., K_{R_L}(\sigma, \tau)) \quad (3)$$

In this paper, we investigate two types of threshold intervals, namely (1) fixed threshold intervals and (2) multiple threshold intervals. For the fixed threshold intervals [38], [39], [34], the spatial (temporal) thresholds follow $\sigma_2 - \sigma_1 = \sigma_3 - \sigma_2 = ... = \sigma_S - \sigma_{S-1}$ ($\tau_2 - \tau_1 = \tau_3 - \tau_2 = ... = \tau_T - \tau_{T-1}$), which are the commonly used way to generate a spatiotemporal K-function plot. For the multiple threshold intervals [63], [64], [65], [66], we do not have any assumption for those spatial and temporal thresholds.

B. Range-Query-based Solution (RQS)

Observe from Equation 1 that only those data points $(\mathbf{p}_j, t_{\mathbf{p}_j})$ that are within the spatial threshold σ and the temporal threshold τ from each data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$ can contribute to the spatiotemporal K-function (see the yellow cylinders in Figures 3b and c). Therefore, one possible approach is to first find the range query set $\mathbb{R}_{\sigma,\tau}(\mathbf{p}_i, t_{\mathbf{p}_i})$ from each data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$, where

$$\mathbb{R}_{\sigma,\tau}(\mathbf{p}_i, t_{\mathbf{p}_i}) = \left\{(\mathbf{p}_j, t_{\mathbf{p}_j}) \in P : \begin{matrix} d(\mathbf{p}_i, \mathbf{p}_j) \leq \sigma, d(t_{\mathbf{p}_i}, t_{\mathbf{p}_j}) \leq \tau \\ j \neq i \end{matrix}\right\} \quad (4)$$

Then, based on $\mathbb{R}_{\sigma,\tau}(\mathbf{p}_i, t_{\mathbf{p}_i})$, we can compute the spatiotemporal K-function based on the following expression.

$$K_P(\sigma, \tau) = \sum_{(\mathbf{p}_i, t_{\mathbf{p}_i}) \in P} |\mathbb{R}_{\sigma,\tau}(\mathbf{p}_i, t_{\mathbf{p}_i})| \quad (5)$$

In practice, many tree-based indexing structures, including kd-tree [41], ball-tree [42], and range-tree [43], have been developed for improving the efficiency of solving the range search problem (i.e., efficiently finding $\mathbb{R}_{\sigma,\tau}(\mathbf{p}_i, t_{\mathbf{p}_i})$), which can also be adopted for efficiently computing $K_P(\sigma, \tau)$ and generating a spatiotemporal K-function plot. Despite this, since those spatial thresholds σ_1, σ_2,..., σ_S and temporal thresholds τ_1, τ_2,..., τ_T can be very large theoretically (e.g., tend to ∞), the cardinality $|\mathbb{R}_{\sigma,\tau}(\mathbf{p}_i, t_{\mathbf{p}_i})|$ can be n (i.e., the same as the size of P) for all (σ, τ)-pairs in the worst case. As such, using this method to compute a spatiotemporal K-function takes $O(n^2)$ time, which indicates that the worst-case time complexity of generating a spatiotemporal K-function plot (see Problem 1) remains in $O(LSTn^2)$.

IV. Distribution-and-Aggregation Solution (DNA)

In this section, we first discuss the main idea of our Distribution-aNd-Aggregation (DNA) solution in Section IV-A. Based on the main idea, we then illustrate the DNA solution with fixed threshold intervals in Section IV-B, which can significantly reduce the worst-case time complexity for generating an exact spatiotemporal K-function plot. Next, we further investigate our DNA solution with multiple threshold intervals in Section IV-C. Lastly, we investigate the space complexity of DNA in Section IV-D.

A. Main Idea of DNA

Recall from Equation 5 that we can efficiently compute a spatiotemporal K-function if we can efficiently compute the cardinality $|\mathbb{R}_{\sigma,\tau}(\mathbf{p}_i, t_{\mathbf{p}_i})|$ for each data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$. Moreover, in order to generate a spatiotemporal K-function plot (see Problem 1), we also need to compute ST cardinalities $|\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})|$ with respect to S spatial thresholds, where $1 \leq u \leq S$, and T temporal thresholds, where $1 \leq v \leq T$, for each location data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$, which can be represented by Figure 5 (e.g., $|\mathbb{R}_{\sigma_4,\tau_3}(\mathbf{p}_i, t_{\mathbf{p}_i})| = 8$ denotes the number of black data points inside the green region).

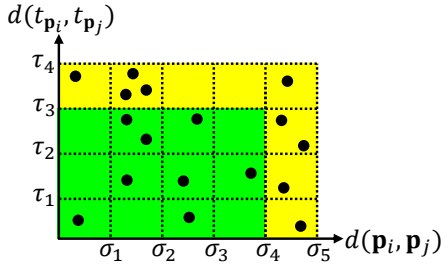


Fig. 5: The main idea of DNA, where the range query set $\mathbb{R}_{\sigma_5,\tau_4}(\mathbf{p}_i, t_{\mathbf{p}_i})$ with the largest spatial threshold σ_5 and the largest temporal threshold τ_4 (i.e., those black data points in the yellow and green regions) contains enough black data points for finding range query sets with smaller spatial and temporal thresholds, e.g., $\mathbb{R}_{\sigma_4,\tau_3}(\mathbf{p}_i, t_{\mathbf{p}_i})$ (see the black data points in the green region).

In Figure 5, observe that the range query set $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ with the largest spatial threshold σ_S and the largest temporal threshold τ_T contains enough black data points for finding $\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})$ with other spatial and temporal thresholds (i.e., σ_u and τ_v, respectively), which indicates that these range query sets exhibit the following coverage property.

$$\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i}) \subseteq \mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i}) \tag{6}$$

where $1 \leq u \leq S$ and $1 \leq v \leq T$.

B. DNA with Fixed Threshold Intervals

In this section, we illustrate the DNA solution, which consists of two steps, namely (1) distribution and (2) aggregation.

Distribution. Recall from the coverage property (see Equation 6) that the range query set $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ with the largest spatial threshold σ_S and the largest temporal threshold τ_T contains enough information for computing $\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})$, where $1 \leq u \leq S$ and $1 \leq v \leq T$. Therefore, instead of computing $\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})$ from scratch for each (σ_u, τ_v)-pair, we first find the range query set $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$. Then, by dividing the $(d(\mathbf{p}_i, \mathbf{p}_j), d(t_{\mathbf{p}_i}, t_{\mathbf{p}_j}))$-plane into $S \times T$ grid cells, we can distribute this set of data points $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ into different grid cells (see Figure 6), where $\mathbb{G}(u, v)$ denotes the number of data points that are inside the grid cell (u, v) (see Equation 7).

$$\mathbb{G}(u,v) = \left| \left\{ (\mathbf{p}_j, t_{\mathbf{p}_j}) \in \mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i}) : \begin{matrix} \sigma_{u-1} < d(\mathbf{p}_i, \mathbf{p}_j) \leq \sigma_u \\ \tau_{v-1} < d(t_{\mathbf{p}_i}, t_{\mathbf{p}_j}) \leq \tau_v \end{matrix} \right\} \right| \tag{7}$$

where we let the dummy variables σ_0 and τ_0 to be 0^-.[2]

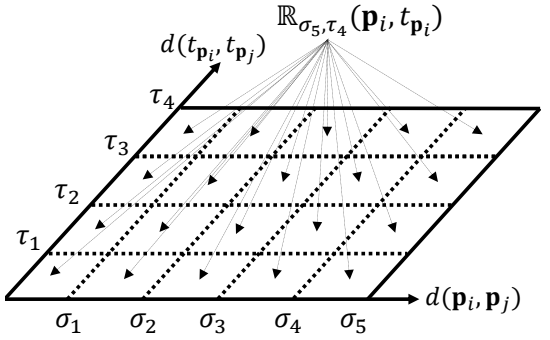


Fig. 6: Once we have obtained the range query set $\mathbb{R}_{\sigma_5,\tau_4}(\mathbf{p}_i, t_{\mathbf{p}_i})$ with the largest spatial threshold σ_5 and the largest temporal threshold τ_4 (i.e., $S = 5$ and $T = 4$, respectively), those data points in $\mathbb{R}_{\sigma_5,\tau_4}(\mathbf{p}_i, t_{\mathbf{p}_i})$ can be distributed to other grid cells (see the black dashed arrows).

The detailed pseudocode can be found in Algorithm 1. In Lemma 1, we further state that the time complexity of this step is $O(n + ST)$ for each data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$.

Algorithm 1 Distribution (Step 1)

1: **procedure** DISTRIBUTION(Data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$, spatial thresholds $\sigma_1, \sigma_2, ..., \sigma_S$, temporal thresholds $\tau_1, \tau_2, ..., \tau_T$)
2: Obtain $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ ▷ Section III-B
3: Initialize the grid $\mathbb{G}$ with $S \times T$ grid cells, i.e., set $\mathbb{G}(u, v) = 0$, where $1 \leq u \leq S$ and $1 \leq v \leq T$
4: **for** each point $(\mathbf{p}_j, t_{\mathbf{p}_j}) \in \mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ **do**
5: Find u such that $\sigma_{u-1} < d(\mathbf{p}_i, \mathbf{p}_j) \leq \sigma_u$
6: Find v such that $\tau_{v-1} < d(t_{\mathbf{p}_i}, t_{\mathbf{p}_j}) \leq \tau_v$
7: $\mathbb{G}(u, v) \leftarrow \mathbb{G}(u, v) + 1$
8: Return $\mathbb{G}$

Lemma 1. *Given a data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$, S spatial thresholds, and T temporal thresholds, the time complexity of the distribution step is $O(n + ST)$.*

Proof. Note that the time complexity of finding the range query set $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ and initializing the grid $\mathbb{G}$ is $O(n)$ (with $\sigma_S \to \infty$ and $\tau_T \to \infty$ in the worst case) and $O(ST)$ time, respectively. Therefore, line 2 and line 3 of

[2]Mathematically, 0^- means a value that is (1) smaller than 0 and (2) arbitrarily close to 0.

Algorithm 1 take $O(n + ST)$ time. In addition, since we generate a spatiotemporal K-function plot with fixed threshold intervals (i.e., $\sigma_2 - \sigma_1 = \sigma_3 - \sigma_2 = ... = \sigma_S - \sigma_{S-1}$ and $\tau_2 - \tau_1 = \tau_3 - \tau_2 = ... = \tau_T - \tau_{T-1}$), this algorithm takes $O(1)$ time for finding the correct positions of u (line 5) and v (line 6) in order to update the number of data points $\mathbb{G}(u, v)$ of the grid cell (u, v). Therefore, the time complexity of line 4 is $O(n)$ throughout all iterations. With the above discussion, we can conclude that the time complexity of this distribution step (i.e., Algorithm 1) is $O(n + ST)$. □

Aggregation. Observe from Figure 5 that computing the value $|\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})|$ is equivalent to aggregating the number of data points that are inside a rectangular region (which covers several grid cells). As an example, finding $|\mathbb{R}_{\sigma_4,\tau_3}(\mathbf{p}_i, t_{\mathbf{p}_i})|$ is equivalent to aggregating all data points inside the green region that consists of 12 grid cells (see Figure 5). Hence, we adopt the idea of [67] and build the prefix count array, $\mathbb{PC}$ (see Equation 8), for the grid $\mathbb{G}$ (in Figure 6) in order to efficiently compute $|\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})|$.

$$\mathbb{PC}(u,v) = \begin{cases} \mathbb{G}(u,v) & \text{if } u = 1, v = 1 \\ \mathbb{PC}(u-1,v) + \mathbb{G}(u,v) & \text{if } u \neq 1, v = 1 \\ \mathbb{PC}(u,v-1) + \mathbb{G}(u,v) & \text{if } u = 1, v \neq 1 \\ \mathbb{PC}(u-1,v) + \mathbb{PC}(u,v-1) & \\ \quad -\mathbb{PC}(u-1,v-1) + \mathbb{G}(u,v) & \text{if } u \neq 1, v \neq 1 \end{cases} \tag{8}$$

Figure 7b shows the prefix count array for the grid $\mathbb{G}$ (see Figure 7a) of Figure 5. Observe that the values 8 and 17 (in the green cell and the yellow cell of Figure 7b) indicate the aggregate values of the green region and the both regions (with yellow and green), respectively, of $\mathbb{G}$ (see Figure 7a). Therefore, we can conclude that $|\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})| = \mathbb{PC}(u, v)$ (see Lemma 2).

Lemma 2. *Given a data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$ and a prefix count array $\mathbb{PC}$ for a grid $\mathbb{G}$, we have*

$$|\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})| = \mathbb{PC}(u, v) \tag{9}$$

Hence, we only need to construct the prefix count array $\mathbb{PC}$ (based on Equation 8)[3] and then scan this array in order to obtain the values $|\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})|$ with all pairs of spatial and temporal thresholds (σ_u, τ_v). The detailed pseudocode can be found in Algorithm 2.

In Lemma 3, we state that the time complexity of this aggregation step is $O(ST)$.

Lemma 3. *Given a data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$, S spatial thresholds, T temporal thresholds, and the grid $\mathbb{G}$, the time complexity of the aggregation step is $O(ST)$.*

Proof. Since computing $\mathbb{PC}(u, v)$ (line 6) and $RA(u, v)$ (line 7) only takes $O(1)$ time, which is based on Equation 8, the time complexity of line 4 to line 7 is $O(ST)$ (with ST iterations in total). Moreover, the initialization of the prefix count array $\mathbb{PC}$ and the result array RA (line 2 and line 3, respectively) also takes $O(ST)$ time. Hence, the time complexity of this aggregation step (i.e., Algorithm 2) is $O(ST)$. □

[3]By adopting the increasing lexicographical order of the variables v and u (i.e., increase v and then increase u), $\mathbb{PC}(u-1, v)$, $\mathbb{PC}(u, v-1)$, and $\mathbb{PC}(u-1, v-1)$ must be available before computing $\mathbb{PC}(u, v)$ in Equation 8.

Algorithm 2 Aggregation (Step 2)

1: **procedure** AGGREGATION(Data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$, spatial thresholds $\sigma_1, \sigma_2, ..., \sigma_S$, temporal thresholds $\tau_1, \tau_2, ..., \tau_T$, Grid $\mathbb{G}$)
2: Initialize the prefix count array $\mathbb{PC}$, with $S \times T$ grid cells, i.e., set $\mathbb{PC}(u, v) = 0$, where $1 \leq u \leq S$ and $1 \leq v \leq T$
3: Initialize the result array RA with $S \times T$ values, where $RA(u, v)$ means $|\mathbb{R}_{\sigma_u,\tau_v}(\mathbf{p}_i, t_{\mathbf{p}_i})|$
4: **for** $u \leftarrow 1$ to S **do**
5: **for** $v \leftarrow 1$ to T **do**
6: Compute $\mathbb{PC}(u, v)$ ▷ Equation 8
7: $RA(u, v) \leftarrow \mathbb{PC}(u, v)$ ▷ Lemma 2
8: Return $RA(u, v)$ for all threshold pairs (u, v)

DNA. With these two steps, distribution and aggregation, for each data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$ in a dataset P, we can compute the spatiotemporal K-function value with respect to each pair of spatial threshold σ_u and temporal threshold τ_v (i.e., ST spatiotemporal K-function values in total) for a dataset P. The detailed pseudocode can be found in Algorithm 3.

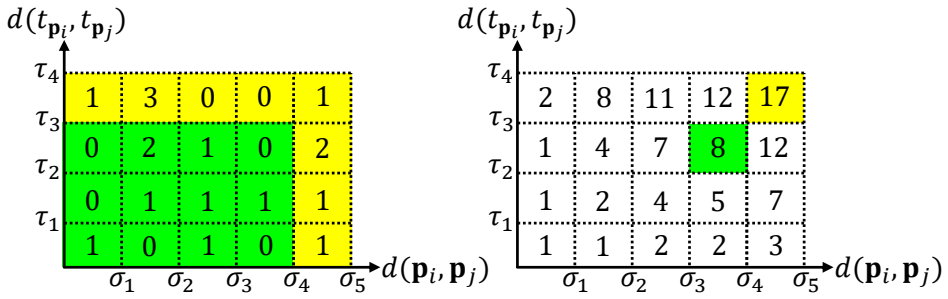
Algorithm 3 DNA

1: **procedure** DNA(Spatiotemporal dataset P, spatial thresholds $\sigma_1, \sigma_2, ..., \sigma_S$, temporal thresholds $\tau_1, \tau_2, ..., \tau_T$)
2: Initialize $\mathbb{K}$ to be an array with size $S \times T$, i.e., set $\mathbb{K}(u, v) = 0$ (which stores the K-function value of P with the spatial threshold σ_u and the temporal threshold τ_v), where $1 \leq u \leq S$ and $1 \leq v \leq T$
3: **for** each $(\mathbf{p}_i, t_{\mathbf{p}_i}) \in P$ **do**
4: //Step 1 (call Algorithm 1)
5: $\mathbb{G} \leftarrow$ DISTRIBUTION$((\mathbf{p}_i, t_{\mathbf{p}_i}), \sigma_1, ..., \sigma_S, \tau_1, ..., \tau_T)$
6: //Step 2 (call Algorithm 2)
7: $RA \leftarrow$ AGGREGATION$((\mathbf{p}_i, t_{\mathbf{p}_i}), \sigma_1, ..., \sigma_S, \tau_1, ..., \tau_T, \mathbb{G})$
8: **for** $1 \leq u \leq S$ **do**
9: **for** $1 \leq v \leq T$ **do**
10: $\mathbb{K}(u, v) \leftarrow \mathbb{K}(u, v) + RA(u, v)$
11: Return $\mathbb{K}$

In Theorem 1, we state that the time complexity of DNA for generating a spatiotemporal K-function plot (i.e., solving Problem 1) is $O(Ln^2 + LSTn)$.

Theorem 1. *The time complexity of using DNA to generate a spatiotemporal K-function plot (see Problem 1) with fixed threshold intervals is $O(Ln^2 + LSTn)$.*

Proof. In Algorithm 3, note that we need to call the distribution method (i.e., Algorithm 1) in line 5 and the aggregation method (i.e., Algorithm 2) in line 7 for each data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$ in the dataset P (i.e., for each iteration of line 3), which take $O(n + ST)$ time (see Lemma 1) and $O(ST)$



(a) Grid $\mathbb{G}$ (b) Prefix count array $\mathbb{PC}$

Fig. 7: The prefix count array $\mathbb{PC}$ (in (b)) is constructed for the grid $\mathbb{G}$ (in (a)) of the example in Figure 5.

time (see Lemma 3), respectively. Furthermore, it also takes $O(ST)$ time for updating the K-function values (i.e., line 8 to line 10). Hence, the time complexity of line 3 to line 10 is $O(n(n+ST)) = O(n^2 + STn)$. In addition, the initialization step in line 2 takes $O(ST)$ time. Therefore, we can conclude that the time complexity of DNA is $O(n^2 + STn)$.

Recall from Problem 1 that we need to compute ST K-function values for $L+1$ datasets in order to generate a spatiotemporal K-function plot. As such, the time complexity of using DNA, by calling it $L+1$ times, for generating a spatiotemporal K-function plot is $O(Ln^2 + LSTn)$. □

Compared with the existing solution, RQS (see Section III-B), which takes $O(LSTn^2)$ time for generating a spatiotemporal K-function plot, DNA can further reduce the time complexity (see Theorem 1) for supporting this tool.

C. DNA with Multiple Threshold Intervals

Although fixed threshold intervals have been frequently adopted for generating a spatiotemporal K-function plot, domain experts [63], [64], [65], [66] can possibly choose multiple threshold intervals for analyzing location datasets in practice. As an example, they can choose 50m, 100m, 200m, 400m, and 800m as the spatial thresholds. Under this setting, each grid cell in Figure 5 does not have the same size. Therefore, after we have obtained the range query set $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ (with the largest spatial and temporal thresholds, σ_S and τ_T, respectively), the distribution step of DNA needs to utilize the binary search method to assign each data point $(\mathbf{p}_j, t_{\mathbf{p}_j})$ (in $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$) into different grid cells (see Figure 6), which incurs the additional time complexity $O(\log S + \log T) = O(\log ST)$ for each $(\mathbf{p}_j, t_{\mathbf{p}_j})$. Therefore, the time complexity of the distribution step becomes $O(n \log ST + ST)$ (instead of $O(n + ST)$ time in Lemma 1). With this overhead, we further conclude that the time complexity of DNA is $O(Ln^2 \log ST + LSTn)$ for generating a spatiotemporal K-function plot with multiple threshold intervals (see Theorem 2). Due to its simplicity, we omit this proof in this paper.

Theorem 2. *The time complexity of using DNA to generate a spatiotemporal K-function plot (see Problem 1) with multiple threshold intervals is $O(Ln^2 \log ST + LSTn)$.*

Although there is an additional $O(\log ST)$ factor in the time complexity of our method, DNA can still theoretically outperform the state-of-the-art RQS method (with $O(LSTn^2)$ time) under this setting.

Additional Improvement. Note that the main bottleneck of using DNA with multiple threshold intervals is that we need to utilize the binary search method in the distribution step. However, those spatial and temporal thresholds may still follow some regular patterns (e.g., $\sigma_u = f(u)$ and $\tau_v = h(v)$) in practice, which can be adopted to get rid of that $O(\log ST)$ factor. As an example, the spatial thresholds, 50m, 100m, 200m, 400m, and 800m, can be represented by a simple function $\sigma_u = 50 \times 2^{u-1}$, where $1 \le u \le 5$. Here, we state that we can find the correct grid cell (u, v) in $O(1)$ time for each data point in $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ (see Figure 6) if (1) the function f and the function h are strictly monotonic increasing and (2) the inverse of these functions, i.e., f^{-1} and h^{-1}, can be computed in $O(1)$ time (see Lemma 4).[4]

Lemma 4. *Given any data point $(\mathbf{p}_j, t_{\mathbf{p}_j})$ in $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$, S spatial thresholds, and T temporal thresholds, where $\sigma_u = f(u)$ $(1 \le u \le S)$ and $\tau_v = h(v)$ $(1 \le v \le T)$, the correct grid cell (u, v) for $(\mathbf{p}_j, t_{\mathbf{p}_j})$ can be found in $O(1)$ time if (1) the function f and the function h are strictly monotonic increasing and (2) the inverse of these functions, i.e., f^{-1} and h^{-1}, can be computed in $O(1)$ time.*

Proof. Suppose that f is strictly monotonic increasing. f^{-1} is also strictly monotonic increasing [68]. Therefore, we can conclude that $u - 1 < f^{-1}(d(\mathbf{p}_i, \mathbf{p}_j)) \le u$ if $\sigma_{u-1} < d(\mathbf{p}_i, \mathbf{p}_j) \le \sigma_u$. Therefore, if f^{-1} can be computed in $O(1)$ time, we can correctly identify that $d(\mathbf{p}_i, \mathbf{p}_j)$ is in the region $(u-1, u]$ in $O(1)$ time. Similarly, we can also easily extend the above proof to the function h. □

As a remark, these two conditions can be easily fulfilled by commonly used functions, e.g., the exponential function (i.e., obtaining a spatiotemporal K-function plot with log scale in the axes of spatial threshold and temporal threshold).

D. Space Complexity of DNA

Recall that generating a spatiotemporal K-function plot needs to access $L+1$ dataset (with size $(L+1)n$) and obtain the spatiotemporal K-function values for ST spatial-temporal threshold-pairs (see Figure 4). The space complexity of every algorithm is at least $O(nL + ST)$.

[4] We can replace line 5 and line 6 of Algorithm 1 by "Set $\lceil f^{-1}(d(\mathbf{p}_i, \mathbf{p}_j)) \rceil$ as u" and "Set $\lceil h^{-1}(d(t_{\mathbf{p}_i}, t_{\mathbf{p}_j})) \rceil$ as v".

Since our method, DNA, needs to call the RQS method for each data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$ with the largest spatial threshold σ_S and the largest temporal threshold τ_T, which takes $O(n)$ auxiliary space, and then distribute these data points of $\mathbb{R}_{\sigma_S,\tau_T}(\mathbf{p}_i, t_{\mathbf{p}_i})$ into ST grid cells (see Figure 6), which takes $O(n + ST)$ auxiliary space. As we can clear the auxiliary space after processing each data point $(\mathbf{p}_i, t_{\mathbf{p}_i})$, the space complexity of DNA remains in $O(nL + ST)$. Based on the above discussion, Theorem 3 formally states the space complexity of DNA.

Theorem 3. *The space complexity of using DNA to generate a spatiotemporal K-function plot is $O(nL + ST)$.*

V. Experimental Evaluation

In this section, we first discuss the experiment settings in Section V-A. Then, we conduct the time efficiency experiments of different methods for generating a spatiotemporal K-function plot with fixed threshold intervals and multiple threshold intervals in Section V-B and Section V-C, respectively. After that, we conduct the space efficiency experiments of different methods for generating a spatiotemporal K-function plot in Section V-D. Lastly, we further conduct a case study for illustrating how to simultaneously adopt this spatiotemporal K-function plot and another point pattern analysis tool, called spatiotemporal kernel density visualization (STKDV) [69], [70], [62], for analyzing COVID-19 cases in the north district of Hong Kong [40] in Section V-E.

A. Experiment Settings

In our experiments, we adopt four large-scale spatiotemporal datasets (see Table III), for testing the efficiency performance of different methods in Table I. The main reason for choosing these datasets is that they cover different categories (of application domains). Specifically, we choose three commonly used tree structures, including kd-tree [41], ball-tree [42], and range-tree [43], for finding the range query set $\mathbb{R}_{\sigma,\tau}(\mathbf{p}_i, t_{\mathbf{p}_i})$ in RQS (see Section III-B), namely RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$, respectively. In addition, the implementation of DNA also chooses the kd-tree for obtaining the range query set. We implemented all these methods in C++ and conducted experiments on an Intel i7 2.9GHz PC with 32GB memory. In this paper, we use the response time (sec) and the memory space (MB) for measuring the time efficiency and space efficiency, respectively, for generating spatiotemporal K-function plots of different methods. As a remark, we omit the results that are more than 259,200 sec (i.e., three days).

TABLE III: Datasets.

Name	n	Category	Ref.
Seattle	236,136	Traffic accidents	[71]
Atlanta	363,861	Crime events	[72]
Los Angeles	559,861	Bicycle mobility	[73]
New York	1,541,284	Traffic accidents	[74]

B. Time Efficiency Experiments: Fixed Threshold Intervals

In this section, we test the efficiency of all methods for generating a spatiotemporal K-function plot with fixed threshold intervals. To conduct our experiments, we adopt $\sigma_u = \frac{2000u}{S}$ and $\tau_v = \frac{14v}{T}$ to choose S spatial thresholds and T temporal thresholds, respectively, with fixed threshold intervals. By default, we set $S = 5$ spatial thresholds (i.e., 400m, 800m, 1200m, 1600m, and 2000m), $T = 5$ temporal thresholds (i.e., 2.8 days, 5.6 days, 8.4 days, 11.2 days, and 14 days), and $L = 1$ random dataset for testing. Here, we show the following four experiments.

Varying the number of data points. In order to investigate how the number of data points affects the response time of each method, we first randomly sample each dataset with four sampling ratios, which are 25%, 50%, 75%, and 100% (original dataset), and then measure the response time of all methods with respect to each sampling ratio. Since DNA (with $O(Ln^2 + LSTn)$ time) can significantly reduce the worst-case time complexity for generating a spatiotemporal K-function plot compared with RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$ (with $O(LSTn^2)$ time), DNA achieves 10.16x to 18.11x speedups over these three state-of-the-art methods (see Figure 8).

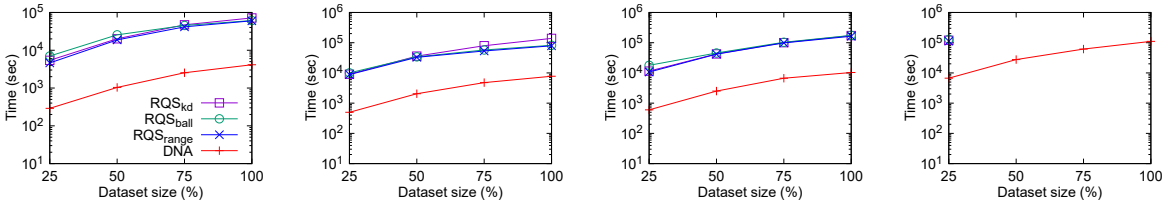
Varying the number of spatial thresholds. We proceed to test how the number of spatial thresholds S affects the time efficiency of all methods. To conduct this experiment, we first choose five values of S, which are 5, 10, 15, 20, and 25, and then test the response time of each method with respect to each S. Figure 9 shows the results of all methods. Observe that the higher the value of S, the longer the response time of RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$. The main reason is that the time complexity of these methods is $O(LSTn^2)$, which is linearly proportional to S. Since the time complexity of DNA is only $O(Ln^2 + LSTn)$ (i.e., it is dominated by the first term), the response time of DNA is not sensitive to S. As such, DNA (1) achieves 10.16x to 55.25x speedups and (2) is more scalable to S compared with RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$.

Varying the number of temporal thresholds. We further examine how the number of temporal thresholds T affects the response time of each method. To conduct this experiment, we first choose five values of T, namely 5, 10, 15, 20, and 25. Then, we measure the response time of each method with respect to each T. Like the discussion for the experiment of varying the number of spatial thresholds, the response time of DNA (1) can achieve 10.16x to 57.42x speedups and (2) is more scalable to T compared with RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$ (see Figure 10).

Varying the number of random datasets. In this experiment, we test how the number of random datasets L affects the time efficiency of all methods. To conduct this experiment, we first choose four numbers of random datasets L, which are 1, 2, 3, and 4, and then test the response time of each method with respect to each L. Figure 11 shows the results of all methods. Observe that the larger the number L, the longer the response time of all methods. The main reason is that the time complexity of each method is linearly proportional to L (see Table I). With the lower time complexity of DNA, it achieves 10.16x to 15.81x speedups over RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$.

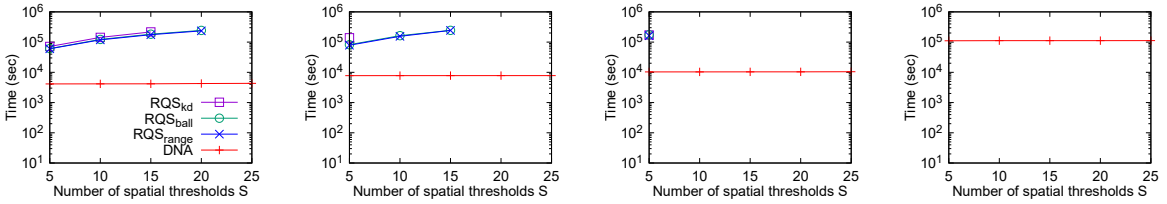
C. Time Efficiency Experiments: Multiple Threshold Intervals

In this section, we investigate the time efficiency of different methods for generating a spatiotemporal K-function plot



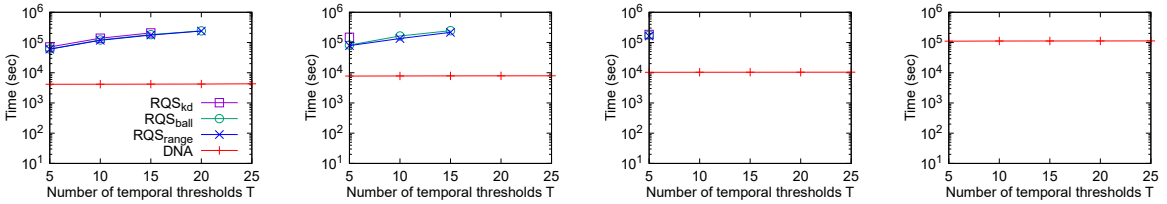
(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 8: Response time for generating a spatiotemporal K-function plot, varying the dataset size n.



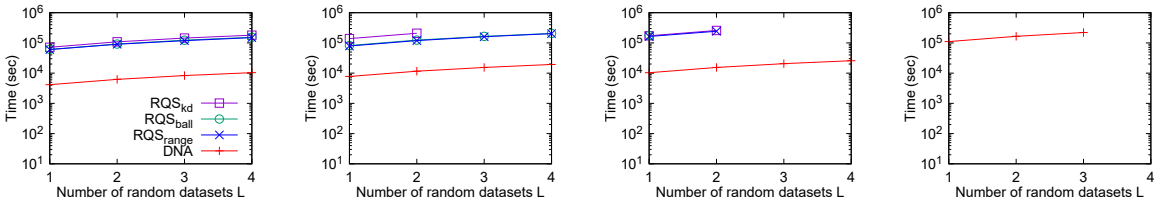
(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 9: Response time for generating a spatiotemporal K-function plot, varying the number of spatial thresholds S.



(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 10: Response time for generating a spatiotemporal K-function plot, varying the number of temporal thresholds T.



(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 11: Response time for generating a spatiotemporal K-function plot, varying the number of random datasets L.

with two types of multiple threshold intervals, which are the (1) random pattern and (2) exponential-function pattern. By default, we also set S, T, and L to be 5, 5 and 1, respectively.

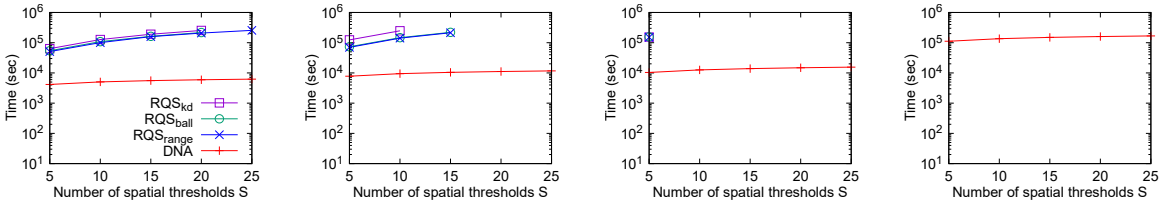
Random pattern. To generate spatial thresholds and temporal thresholds based on random pattern, we randomly choose S and T thresholds within the range from 0 to 2000m and within the range from 0 to 14 days, respectively.

In the first experiment, we choose five values of S, which are 5, 10, 15, 20, and 25, for testing the response time of each method with respect to each S. Figure 12 shows the results of all methods. Although DNA suffers from additional $O(\log ST)$ time overhead compared with the one with fixed threshold intervals, this method can still achieve 9.01x to 41.1x speedups compared with RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$.

In the second experiment, we measure the response time of all methods in each dataset by varying T (from 5 to 25). Figure 13 shows the results of all methods. Observe that the larger the parameter T, the larger the time gap between RQS_{kd}/RQS_{ball}/$\text{RQS}_{\text{range}}$ and DNA. Note that DNA can outperform these existing methods by 9.01x to 35.69x no matter which T we adopt.

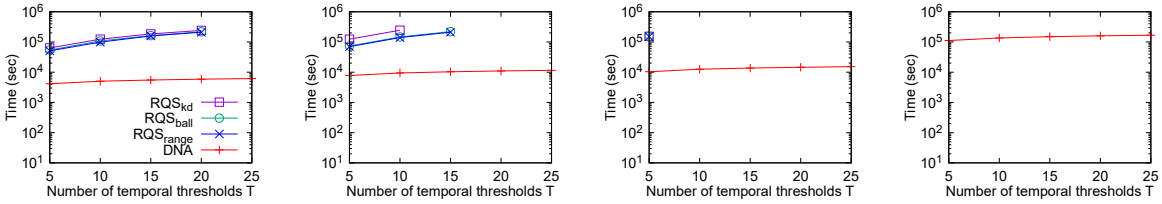
Exponential-function pattern. In this experiment, we adopt the exponential functions, $2000 \times 2^{u-S}$ and $14 \times 2^{v-T}$, for obtaining the sequence of S spatial thresholds and the sequence of T temporal thresholds, respectively. Using $S = 5$ and $T = 5$ as an example, the spatiotemporal K-function plot contains five spatial thresholds with 125m, 250m, 500m, 1000m, and 2000m and five temporal thresholds with 0.875 days, 1.75 days, 3.5 days, 7 days, and 14 days.

In the first experiment, we choose five values of S, which are 2, 3, 4, 5, and 6, for testing. Observe from Figure 14 that the response time of DNA and DNA_{AI} (using the additional improvement approach that is stated in Section IV-C) is not sensitive to this parameter S. The main reason is that the dominant term of the time complexity of DNA/DNA_{AI} only logarithmically increases/does not increase with respect to S. Due to the low time complexity of DNA and DNA_{AI}, these two methods can achieve 4.58x to 45.93x speedups compared with RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$. As a remark, DNA_{AI} achieves 1.18x to 2.15x speedups over DNA.



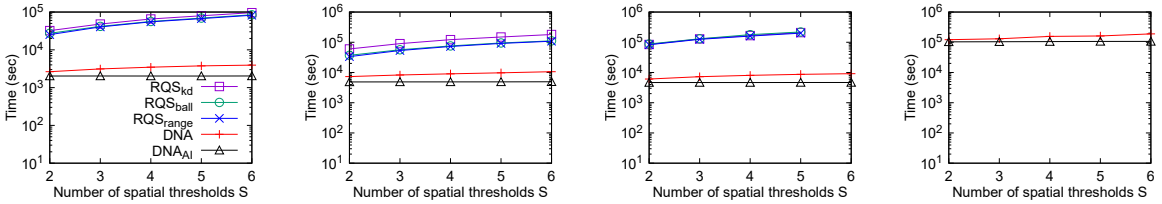
(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 12: Response time for generating a spatiotemporal K-function plot, varying the number of spatial thresholds S (following the random pattern).



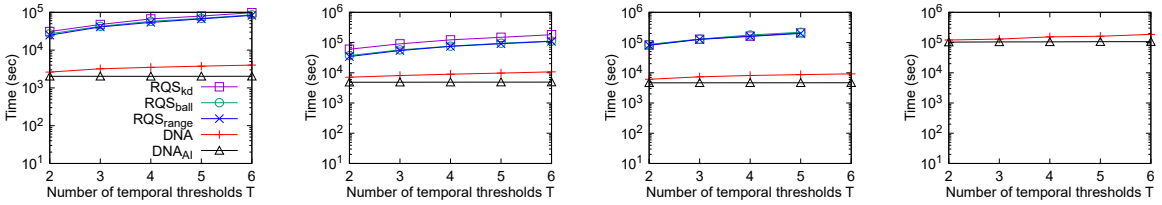
(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 13: Response time for generating a spatiotemporal K-function plot, varying the number of temporal thresholds T (following the random pattern).



(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 14: Response time for generating a spatiotemporal K-function plot, varying the number of spatial thresholds S (following the exponential-function pattern).



(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

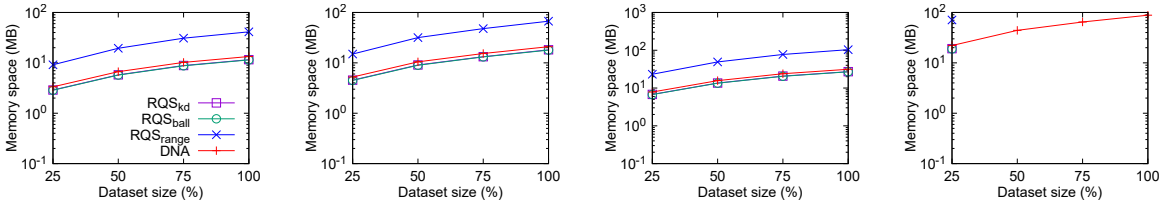
Fig. 15: Response time for generating a spatiotemporal K-function plot, varying the number of temporal thresholds T (following the exponential-function pattern).

In the second experiment, we choose five values of T, which are 2, 3, 4, 5, and 6, for testing. Figure 15 shows the results of all methods. Due to the similar reason as the first experiment, DNA and DNA_{AI} can outperform RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$ by 4.83x to 42.94x. Furthermore, DNA_{AI} achieves 1.18x to 2.18x speedups over DNA.

D. Space Efficiency Experiments

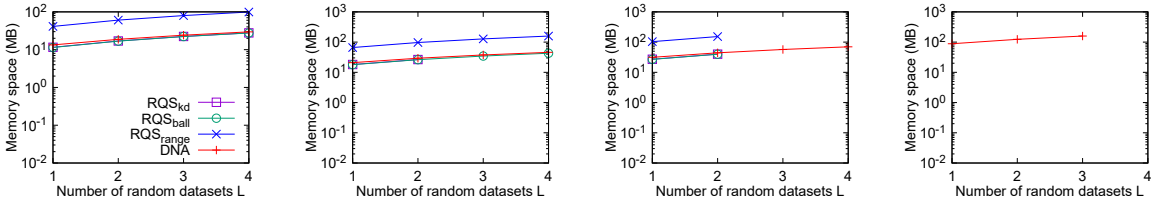
We further investigate the space efficiency of each method for generating a spatiotemporal K-function plot with fixed threshold intervals by conducting the following experiments, which are (1) varying the number of data points, (2) varying the number of random datasets L, (3) varying the number of spatial thresholds, and (4) varying the number of temporal thresholds. By default, we choose the number of spatial thresholds S, the number of temporal thresholds T, and the number of random datasets L to be 5, 5, and 1, respectively (i.e., follow the same default settings as Section V-B). As a remark, we omit those results which take more than three days (i.e., 259,200 sec) to run. Due to space limitations, we only adopt the Seattle and Atlanta datasets for the experiments (3) and (4).

Varying the number of data points. In this experiment, we first sample each dataset with four sampling ratios, which are 25%, 50%, 75%, and 100% (original one). Then, we measure the space consumption of each method in these reduced datasets. Observe from Figure 16 that the memory space consumption of all methods, including RQS_{kd}, RQS_{ball}, $\text{RQS}_{\text{range}}$, and DNA, is proportional to the dataset size n since the space complexity of all these methods is $O(nL + ST)$ (see Table I). Note that $\text{RQS}_{\text{range}}$ takes comparatively larger memory space compared with other methods since this method needs to construct multiple tree structures.



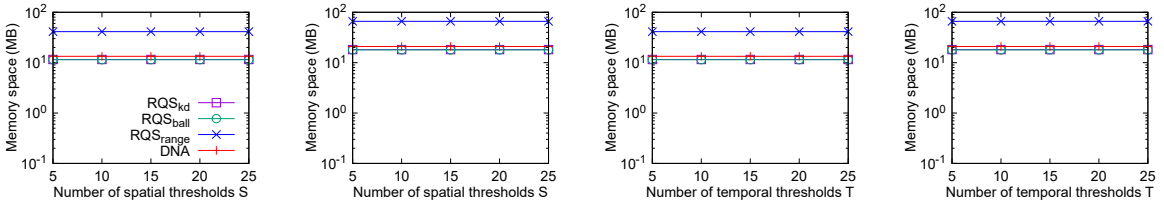
(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 16: Memory space consumption for generating a spatiotemporal K-function plot, varying the dataset size n.



(a) Seattle (b) Atlanta (c) Los Angeles (d) New York

Fig. 17: Memory space consumption for generating a spatiotemporal K-function plot, varying the number of random datasets L.



(a) Seattle (b) Atlanta (c) Seattle (d) Atlanta

Fig. 18: Memory space consumption for generating a spatiotemporal K-function plot, varying the number of spatial thresholds S (a and b) and the number of temporal thresholds T (c and d).

Varying the number of random datasets. We proceed to investigate how the number of random datasets L affects the memory space consumption of all methods. To conduct this experiment, we choose L to be 1, 2, 3, and 4 for testing. Figure 17 shows the results of all methods. Since the space complexity of all these methods is $O(nL + ST)$ (where nL is normally much larger than ST in practice), the memory space consumption of all these methods is proportional to this parameter L. With the same space complexity of DNA, RQS_{kd}, RQS_{ball}, and $\text{RQS}_{\text{range}}$, these methods have similar memory space consumption no matter which L we choose.

Varying the number of spatial thresholds. In this experiment, we test how the number of spatial thresholds S affects the memory space consumption of each method by choosing five values of S, which are 5, 10, 15, 20, and 25. Observe from Figures 18a and b that the memory space consumption of each method is not sensitive to S and is similar to each other. The main reason is that the space complexity of each method is $O(nL + ST)$ (see Table I) and n is much larger than ST.

Varying the number of temporal thresholds. Here, we also choose five values of T, which are 5, 10, 15, 20, and 25, for testing the memory space consumption of each method. Based on the similar reason, we note that the memory space consumption of each method is also not sensitive to T and is similar to each other (see Figures 18c and d).

Further discussion. We do not test the practical space consumption for generating a spatiotemporal K-function plot with multiple threshold intervals. However, we expect the results to be similar to previous experiments because DNA only uses the binary search (or the functions f^{-1} and h^{-1} in Lemma 4) in the distribution algorithm. This incurs no space overhead.

E. Case Study

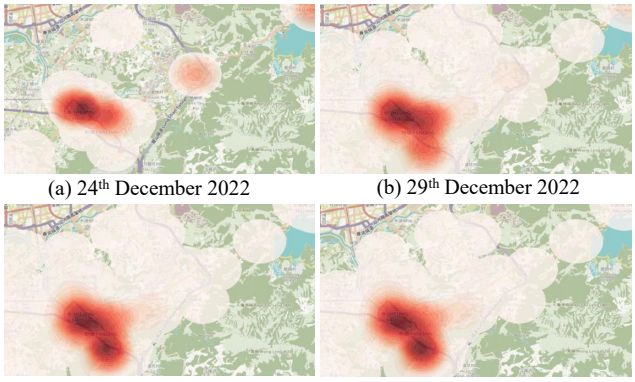
In practice, domain experts simultaneously adopt the spatiotemporal K-function plot and other point pattern analysis tools, e.g., clustering [39], [35] and hotspot detection tools [69], [37], [75], for analyzing location datasets. In this section, we illustrate (1) how to understand the implication of COVID-19 cases in the north district of Hong Kong based on the spatiotemporal K-function plot (see Figure 4), (2) how to use spatiotemporal kernel density visualization (STKDV), which is a hotspot detection tool, with the spatiotemporal K-function plot to identify meaningful (or significant) hotspots of COVID-19 cases in the north district of Hong Kong, and (3) how DNA can benefit this analysis task.

Understand the implication of COVID-19 cases based on the spatiotemporal K-function plot. Recall from Figure 4 that this is the spatiotemporal K-function plot of the COVID-19 cases in the north district of Hong Kong. Observe that those COVID-19 cases tend to have significant clusters if we adopt small spatial thresholds and small temporal thresholds. Once the spatial threshold is set to be very large (e.g., $>$ 10000m), those COVID-19 cases tend to be random/dispersed, especially for relatively large temporal thresholds (e.g., more than 8 days). Furthermore, another observation is that the larger the temporal thresholds, the higher the possibility for those COVID-19 cases to be random/dispersed (under the same

spatial threshold). Therefore, we should avoid discovering a hotspot/cluster with a large spatial threshold and a large temporal threshold simultaneously (i.e., the ranges where the green surface is below the red and blue surfaces in Figure 4). ***Identify meaningful (or significant) hotspots of COVID-19 cases.*** Here, we adopt STKDV to identify spatiotemporal hotspots of the COVID-19 cases in the north district of Hong Kong (see Figure 19). To generate these hotspot maps, we color each pixel-timestamp pair (i.e., $(\mathbf{q}, t_i)$-pair) based on the following spatiotemporal kernel density function $\mathcal{F}_P(\mathbf{q}, t_i)$ (with the Epanechnikov kernel function).

$$\mathcal{F}_P(\mathbf{q}, t_i) = \frac{1}{|P|} \sum_{(\mathbf{p}, t_\mathbf{p}) \in P} \begin{cases} 1 - \frac{1}{b_{\text{space}}^2} d(\mathbf{q}, \mathbf{p})^2 & \text{if } d(\mathbf{q}, \mathbf{p}) \leq b_{\text{space}} \\ 0 & \text{otherwise} \end{cases} \cdot \begin{cases} 1 - \frac{1}{b_{\text{time}}^2} d(t_i, t_\mathbf{p})^2 & \text{if } d(t_i, t_\mathbf{p}) \leq b_{\text{time}} \\ 0 & \text{otherwise} \end{cases}$$

Note that the spatial bandwidth b_{space} and the temporal bandwidth b_{time} can significantly affect the hotspot maps. Therefore, there are two challenges. First, it can be challenging to set the correct values for these two parameters. Second, although we have discovered high-density regions in hotspot maps, it is still unknown whether they are significant/meaningful.



(a) 24th December 2022 (b) 29th December 2022

(c) 3rd January 2023 (d) 8th January 2023

Fig. 19: Using STKDV to generate the spatiotemporal hotspot maps using the spatial bandwidth $b_{\text{space}} = 1000$m and the temporal bandwidth $b_{\text{time}} = 4$ days.

To tackle the above challenges, domain experts [37], [75] normally utilize the spatiotemporal K-function plot. Observe from Figure 4 that we should restrict b_{space} and b_{time} to be a spatial threshold and a temporal threshold, respectively, so that the green surface is above the blue surface, which implies that these parameters can likely detect meaningful hotspots. Hence, we choose b_{space} to be 1000m and b_{time} to be 4 days for generating the hotspot maps in Figure 19. Furthermore, we note that the sizes (e.g., the maximum lengths) of these hotspots (red regions) are normally smaller than 5km. Therefore, based on the spatiotemporal K-function plot (see Figure 4), we can conclude that these hotspot regions are significant/meaningful. ***Adopt DNA for benefiting this analysis task.*** We proceed to investigate the time efficiency for generating this spatiotemporal K-function plot (with $L = 2$ random datasets, $S = 60$ spatial thresholds, and $T = 20$ temporal thresholds) for those COVID-19 cases. Table IV shows the results of all methods.

TABLE IV: Response time (sec) of each method for generating the spatiotemporal K-function plot in the COVID-19 cases.

Method	RQS_{kd}	RQS_{ball}	$\text{RQS}_{\text{range}}$	DNA
Time (seconds)	141679	130385	122195	**921.86**

Observe that DNA can further achieve at least 132.55x speedups for generating this plot compared with existing methods. Therefore, instead of waiting for more than one day to obtain this plot, domain experts only need to wait for roughly 921.86 seconds (i.e., 15.36 minutes) by adopting DNA. As such, they can perform detailed analysis by generating more spatiotemporal K-function plots for those data points in different regions of Hong Kong (e.g., Kwun Tong district and Mong Kok District).

VI. Conclusion

In this paper, we study the problem of generating a spatiotemporal K-function plot, which is an important point pattern analysis tool that has been frequently used in a wide range of applications, e.g., criminology, transportation science, urban planning, and epidemiology. However, the state-of-the-art solution, i.e., RQS, suffers from high time complexity (with $O(LSTn^2)$ time) for supporting this tool, which has been widely complained by many domain experts. In order to improve the efficiency for supporting this tool, we develop the first solution, called Distribution-aNd-Aggregation (DNA), which successfully reduces the worst-case time complexity of generating a spatiotemporal K-function plot with fixed threshold intervals to $O(Ln^2 + LSTn)$. Furthermore, by slightly modifying DNA, we show that this method can generate this plot with multiple threshold intervals in $O(Ln^2 \log ST + LSTn)$ time. Hence, DNA is theoretically more scalable to generate this plot with multiple spatial and temporal thresholds. In addition, DNA also retains the same space complexity. Experiment results on four large-scale datasets verify that DNA can achieve speedups of 4.58x to 57.42x over the state-of-the-art methods.

However, the time complexity of DNA (1) still depends on n^2 in the first term and (2) is still theoretically far from optimal.[5] Based on these reasons, DNA can still take more than 10^5 seconds for handling million-scale datasets (e.g., New York in Figure 9d). In the future, we will investigate how to develop an approximate solution (e.g., sampling) to further reduce the time complexity for supporting this tool with a non-trivial accuracy guarantee. Moreover, we will exploit parallel/distributed/modern-hardware-based approaches (e.g., GPU) to further improve the performance of our DNA solution. In addition, we will investigate whether we can progressively and accurately output partial spatiotemporal K-function plots to domain experts so that they can efficiently obtain initial insights for those datasets.

[5]In order to generate a spatiotemporal K-function plot, we need to access all data points in $L + 1$ datasets and compute $(L + 1)ST$ spatiotemporal K-functions (see Problem 1), which results in $\Omega(Ln + LST)$ as the lower bound time complexity for this tool. As such, there is still a substantial time gap compared with DNA (with $O(Ln^2 + LSTn)$ time).

AI-Generated Content Acknowledgement

We do not use any AI tools (e.g., ChatGPT and DeepSeek) to write this article.

References

[1] A. Baddeley, E. Rubak, and R. Turner, *Spatial Point Patterns: Methodology and Applications with R*. London: Chapman and Hall/CRC Press, 2015. [Online]. Available: https://www.routledge.com/Spatial-Point-Patterns-Methodology-and-Applications-with-R/Baddeley-Rubak-Turner/9781482210200/

[2] P. J. Diggle, *Statistical analysis of spatial and spatio-temporal point patterns*, 3rd ed. Boca Raton: CRC Press, 2014.

[3] B. D. Ripley, "The second-order analysis of stationary point processes," *Journal of Applied Probability*, vol. 13, no. 2, p. 255–266, 1976.

[4] M. Liang, H. Li, R. W. Liu, J. S. L. Lam, and Z. Yang, "PiracyAnalyzer: Spatial temporal patterns analysis of global piracy incidents," *Reliability Engineering & System Safety*, vol. 243, p. 109877, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832023007913

[5] P. J. Brantingham, J. Carter, J. MacDonald, C. Melde, and G. Mohler, "Is the recent surge in violence in American cities due to contagion?" *Journal of Criminal Justice*, vol. 76, p. 101848, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0047235221000684

[6] E. García-Tejeda, G. Fondevila, and O. S. Siordia, "Spatial analysis of gunshot reports on twitter in Mexico City," *ISPRS International Journal of Geo-Information*, vol. 10, no. 8, 2021. [Online]. Available: https://www.mdpi.com/2220-9964/10/8/540

[7] P.-F. Kuo and D. Lord, "A promising example of smart policing: A cross-national study of the effectiveness of a data-driven approach to crime and traffic safety," *Case Studies on Transport Policy*, vol. 7, no. 4, pp. 761–771, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213624X19301336

[8] Y. Shi, J. Gong, M. Deng, X. Yang, and F. Xu, "A graph-based approach for detecting spatial cross-outliers from two types of spatial point events," *Computers, Environment and Urban Systems*, vol. 72, pp. 88–103, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S019897151730100X

[9] X. Chen, L. Huang, D. Dai, M. Zhu, and K. Jin, "Hotspots of road traffic crashes in a redeveloping area of Shanghai," *International journal of injury control and safety promotion*, vol. 25, no. 3, pp. 293–302, 2018.

[10] G. Mountrakis and K. Gunson, "Multi-scale spatiotemporal analyses of moose-vehicle collisions: a case study in northern Vermont," *Int. J. Geogr. Inf. Sci.*, vol. 23, no. 11, pp. 1389–1412, 2009. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/13658810802406132

[11] Y. Sadahiro, "Event pattern analysis: the point density around the appearance and disappearance of points," *Journal of Spatial Science*, vol. 69, no. 2, pp. 649–663, 2024.

[12] J. Vukomanovic, J. B. Vogler, and A. Petrasova, "Modeling the connection between viewscapes and home locations in a rapidly exurbanizing region," *Computers, Environment and Urban Systems*, vol. 78, p. 101388, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0198971519301875

[13] F. Guo, Z. Su, G. Wang, L. Sun, F. Lin, and A. Liu, "Wildfire ignition in the forests of southeast China: Identifying drivers and spatial distribution to predict wildfire likelihood," *Applied Geography*, vol. 66, pp. 12–21, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0143622815300199

[14] E. Marcon and F. Puech, "Evaluating the geographic concentration of industries using distance-based methods," *Journal of economic geography*, vol. 3, no. 4, pp. 409–428, 2003.

[15] S. U. Morton, C. Hehnly, K. Burgoine, P. Ssentongo, J. E. Ericson, M. S. Kumar, C. Hagmann, C. Fronterre, J. Smith, M. Movassagh *et al.*, "Paenibacillus spp infection among infants with postinfectious hydrocephalus in Uganda: an observational case-control study," *The Lancet Microbe*, vol. 4, no. 8, pp. e601–e611, 2023.

[16] S. Zhang, M. Wang, Z. Yang, and B. Zhang, "Do spatiotemporal units matter for exploring the microgeographies of epidemics?" *Applied Geography*, vol. 142, p. 102692, 2022.

[17] Y. Chen, S. Liu, X. Shan, H. Wang, B. Li, J. Yang, L. Dai, J. Liu, and G. Li, "Schistosoma japonicum-infected sentinel mice: surveillance and spatial point pattern analysis in Hubei province, China, 2010–2018," *International Journal of Infectious Diseases*, vol. 99, pp. 179–185, 2020.

[18] C. Zheng, J. Fu, Z. Li, G. Lin, D. Jiang, and X.-n. Zhou, "Spatiotemporal variation and hot spot detection of visceral leishmaniasis disease in Kashi Prefecture, China," *International journal of environmental research and public health*, vol. 15, no. 12, p. 2784, 2018.

[19] X. Liu, S. Komladzei, and C. Guy, "KCBC – a correlation-based method for co-localization analysis of super-resolution microscopy images using bivariate Ripley's k functions," *Journal of Applied Statistics*, pp. 1–17, 2024.

[20] A. Spark, A. Kitching, D. Esteban-Ferrer, A. Handa, A. R. Carr, L.-M. Needham, A. Ponjavic, A. M. Santos, J. McColl, C. Leterrier *et al.*, "vLUME: 3d virtual reality for single-molecule localization microscopy," *Nature Methods*, vol. 17, no. 11, pp. 1097–1099, 2020.

[21] X. Lu, P. R. Nicovich, M. Zhao, D. J. Nieves, M. Mollazade, S. Vivekchand, K. Gaus, and J. J. Gooding, "Monolayer surface chemistry enables 2-colour single molecule localisation microscopy of adhesive ligands and adhesion proteins," *Nature communications*, vol. 9, no. 1, pp. 1–10, 2018.

[22] Q. Gu, W. Nanney, H. H. Cao, H. Wang, and T. Ye, "Single molecule profiling of molecular recognition at a model electrochemical biosensor," *Journal of the American Chemical Society*, vol. 140, no. 43, pp. 14 134–14 143, 2018.

[23] J. Griffié, M. Shannon, C. L. Bromley, L. Boelen, G. L. Burn, D. J. Williamson, N. A. Heard, A. P. Cope, D. M. Owen, and P. Rubin-Delanchy, "A Bayesian cluster analysis method for single-molecule localization microscopy data," *Nature Protocols*, vol. 11, no. 12, pp. 2499–2514, 2016.

[24] P. Rubin-Delanchy, G. L. Burn, J. Griffié, D. J. Williamson, N. A. Heard, A. P. Cope, and D. M. Owen, "Bayesian cluster identification in single-molecule localization microscopy data," *Nature methods*, vol. 12, no. 11, pp. 1072–1076, 2015.

[25] E. G. Healey, B. Bishop, J. Elegheert, C. H. Bell, S. Padilla-Parra, and C. Siebold, "Repulsive guidance molecule is a structural bridge between neogenin and bone morphogenetic protein," *Nature structural & molecular biology*, vol. 22, no. 6, pp. 458–465, 2015.

[26] J. Rossy, D. M. Owen, D. J. Williamson, Z. Yang, and K. Gaus, "Conformational states of the kinase Lck regulate clustering in early T cell signaling," *Nature immunology*, vol. 14, no. 1, pp. 82–89, 2013.

[27] D. J. Williamson, D. M. Owen, J. Rossy, A. Magenau, M. Wehrmann, J. J. Gooding, and K. Gaus, "Pre-existing clusters of the adaptor Lat do not participate in early T cell signaling events," *Nature immunology*, vol. 12, no. 7, pp. 655–662, 2011.

[28] "ArcGIS," https://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-statistics-toolbox/h-how-multi-distance-spatial-cluster-analysis-ripl.htm.

[29] "CrimeStat: Spatial statistics program for the analysis of crime incident locations," https://nij.ojp.gov/topics/articles/crimestat-spatial-statistics-program-analysis-crime-incident-locations.

[30] "spatstat," https://cran.r-project.org/web/packages/spatstat/index.html.

[31] S. J. Rey, L. Anselin, X. Li, R. Pahle, J. Laura, W. Li, and J. Koschinsky, "Open geospatial analytics with PySAL," *ISPRS International Journal of Geo-Information*, vol. 4, no. 2, pp. 815–836, 2015.

[32] S. J. Salyer, J. Maeda, S. Sembuche, Y. Kebede, A. Tshangela, M. Moussif, C. Ihekweazu, N. Mayet, E. Abate, A. O. Ouma *et al.*, "The first and second waves of the COVID-19 pandemic in Africa: a cross-sectional study," *The lancet*, vol. 397, no. 10281, pp. 1265–1275, 2021.

[33] J. Huang, M.-P. Kwan, and Z. Kan, "The superspreading places of COVID-19 and the associated built-environment and socio-demographic features: A study using a spatial network framework and individual-level activity data," *Health & Place*, vol. 72, p. 102694, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1353829221001908

[34] I. Fuentes-Santos, W. González-Manteiga, and J. Zubelli, "Nonparametric spatiotemporal analysis of violent crime. a case study in the Rio de Janeiro metropolitan area," *Spatial Statistics*, vol. 42, p. 100431, 2021, towards Spatial Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211675320300257

[35] A. Wooditch and D. Weisburd, "Using space–time analysis to evaluate criminal justice programs: An application to stop-question-frisk practices," *Journal of quantitative criminology*, vol. 32, pp. 191–213, 2016.

[36] X. Ye, X. Xu, J. Lee, X. Zhu, and L. Wu, "Space–time interaction of residential burglaries in Wuhan, China," *Applied Geography*, vol. 60, pp. 210–216, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014362281400277X

[37] A. Hohl, E. Delmelle, W. Tang, and I. Casas, "Accelerating the discovery of space-time patterns of infectious diseases using parallel computing," *Spatial and Spatio-temporal Epidemiology*, vol. 19, pp. 10–20, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187758451530040X

[38] J. Lee, *Spatiotemporal Analytics*. CRC Press, 2023.

[39] X. Yan, T. Pei, C. Song, and X. Liu, "Estimating spatiotemporal aggregation scales by revisiting the spatiotemporal L-function," *Transactions in GIS*, vol. 27, no. 2, pp. 592–604, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.13034

[40] "Common Spatial Data Infrastructure (CSDI) Portal for Government," https://portal.csdi.gov.hk/csdi-webpage/.

[41] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[42] A. W. Moore, "The anchors hierarchy: Using the triangle inequality to survive high dimensional data," in *UAI*, 2000, pp. 397–405.

[43] T. M. Chan, K. G. Larsen, and M. Pătraşcu, "Orthogonal range searching on the ram, revisited," in *SCG*. ACM, 2011, pp. 1–10. [Online]. Available: https://doi.org/10.1145/1998196.1998198

[44] E. Delmelle, E. C. Delmelle, I. Casas, and T. Barto, "HELP: a GIS-based health exploratory analysis tool for practitioners," *Applied Spatial Analysis and Policy*, vol. 4, pp. 113–137, 2011.

[45] T. N. Chan, L. H. U, Y. Peng, B. Choi, and J. Xu, "Fast network k-function-based spatial analysis," *Proc. VLDB Endow.*, vol. 15, no. 11, pp. 2853–2866, 2022. [Online]. Available: https://www.vldb.org/pvldb/vol15/p2853-chan.pdf

[46] Y. Wang, Z. Gui, H. Wu, D. Peng, J. Wu, and Z. Cui, "Optimizing and accelerating space–time Ripley's k function based on Apache Spark for distributed spatiotemporal point pattern analysis," *Future Generation Computer Systems*, vol. 105, pp. 96–118, 2020.

[47] S. Rakshit, A. Baddeley, and G. Nair, "Efficient code for second order analysis of events on a linear network," *Journal of Statistical Software, Articles*, vol. 90, no. 1, pp. 1–37, 2019. [Online]. Available: https://www.jstatsoft.org/v090/i01

[48] G. Zhang, Q. Huang, A. Zhu, and J. H. Keel, "Enabling point pattern analysis on spatial big data using cloud computing: optimizing and accelerating Ripley's K function," *Int. J. Geogr. Inf. Sci.*, vol. 30, no. 11, pp. 2230–2252, 2016. [Online]. Available: https://doi.org/10.1080/13658816.2016.1170836

[49] W. Tang, W. Feng, and M. Jia, "Massively parallel spatial point pattern analysis: Ripley's K function accelerated using graphics processing units," *Int. J. Geogr. Inf. Sci.*, vol. 29, no. 3, pp. 412–439, 2015. [Online]. Available: https://doi.org/10.1080/13658816.2014.976569

[50] T. N. Chan, L. H. U, B. Choi, and J. Xu, "SLAM: efficient sweep line algorithms for kernel density visualization," in *SIGMOD*. ACM, 2022, pp. 2120–2134. [Online]. Available: https://doi.org/10.1145/3514221.3517823

[51] T. N. Chan, R. Cheng, and M. L. Yiu, "QUAD: quadratic-bound-based kernel density visualization," in *SIGMOD*, 2020, pp. 35–50. [Online]. Available: https://doi.org/10.1145/3318464.3380561

[52] J. M. Phillips and W. M. Tai, "Near-optimal coresets of kernel density estimates," in *SOCG*, 2018, pp. 66:1–66:13. [Online]. Available: https://doi.org/10.4230/LIPIcs.SoCG.2018.66

[53] ——, "Improved coresets for kernel density estimates," in *SODA*, 2018, pp. 2718–2727. [Online]. Available: https://doi.org/10.1137/1.9781611975031.173

[54] E. Gan and P. Bailis, "Scalable kernel density classification via threshold-based pruning," in *ACM SIGMOD*, 2017, pp. 945–959. [Online]. Available: https://doi.org/10.1145/3035918.3064035

[55] Y. Zheng and J. M. Phillips, "L∞ error and bandwidth selection for kernel density estimates of large data," in *SIGKDD*, 2015, pp. 1533–1542. [Online]. Available: http://doi.acm.org/10.1145/2783258.2783357

[56] Y. Zheng, J. Jestes, J. M. Phillips, and F. Li, "Quality and efficiency for kernel density estimates in large data," in *SIGMOD*, 2013, pp. 433–444.

[57] J. M. Phillips, "ϵ-samples for kernels," in *SODA*, 2013, pp. 1622–1632. [Online]. Available: https://doi.org/10.1137/1.9781611973105.116

[58] A. G. Gray and A. W. Moore, "Rapid evaluation of multiple density models," in *AISTATS*. Society for Artificial Intelligence and Statistics, 2003. [Online]. Available: https://proceedings.mlr.press/r4/gray03a.html

[59] ——, "Nonparametric density estimation: Toward computational tractability," in *SDM*, 2003, pp. 203–211. [Online]. Available: https://doi.org/10.1137/1.9781611972733.19

[60] T. N. Chan, R. Zang, B. Zhu, L. H. U, D. Wu, and J. Xu, "LION: fast and high-resolution network kernel density visualization," *Proc. VLDB Endow.*, vol. 17, no. 6, pp. 1255–1268, 2024. [Online]. Available: https://www.vldb.org/pvldb/vol17/p1255-chan.pdf

[61] T. N. Chan, Z. Li, L. H. U, J. Xu, and R. Cheng, "Fast augmentation algorithms for network kernel density visualization," *Proc. VLDB Endow.*, vol. 14, no. 9, pp. 1503–1516, 2021. [Online]. Available: http://www.vldb.org/pvldb/vol14/p1503-chan.pdf

[62] T. N. Chan, P. L. Ip, L. H. U, B. Choi, and J. Xu, "SWS: A complexity-optimized solution for spatial-temporal kernel density visualization," *Proc. VLDB Endow.*, vol. 15, no. 4, pp. 814–827, 2021. [Online]. Available: https://www.vldb.org/pvldb/vol15/p814-chan.pdf

[63] S. Feng, A. Wang, Z. Tian, and S. Park, "Exploring the correlation between hard-braking events and traffic crashes in regional transportation networks: A geospatial perspective," *Multimodal Transportation*, vol. 3, no. 2, p. 100128, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772586324000091

[64] P. Hennerdal and M. M. Nielsen, "A multiscalar approach for identifying clusters and segregation patterns that avoids the modifiable areal unit problem," *Annals of the American Association of Geographers*, vol. 107, no. 3, pp. 555–574, 2017. [Online]. Available: https://doi.org/10.1080/24694452.2016.1261685

[65] M. G. Eberhart, B. R. Yehia, A. Hillier, C. D. Voytek, M. B. Blank, I. Frank, D. S. Metzger, and K. A. Brady, "Behind the cascade: analyzing spatial patterns along the hiv care continuum," *JAIDS Journal of Acquired Immune Deficiency Syndromes*, vol. 64, pp. S42–S51, 2013.

[66] H. X. John R. Logan, Seth Spielman and P. N. Klein, "Identifying and bounding ethnic neighborhoods," *Urban Geography*, vol. 32, no. 3, pp. 334–359, 2011. [Online]. Available: https://doi.org/10.2747/0272-3638.32.3.334

[67] C. Ho, R. Agrawal, N. Megiddo, and R. Srikant, "Range queries in OLAP data cubes," in *SIGMOD*, J. Peckham, Ed., 1997, pp. 73–88. [Online]. Available: https://doi.org/10.1145/253260.253274

[68] K. G. Binmore, *Mathematical Analysis: a straightforward approach*. Cambridge University Press, 1982.

[69] T. N. Chan, P. L. Ip, B. Zhu, L. H. U, D. Wu, J. Xu, and C. S. Jensen, "Large-scale spatiotemporal kernel density visualization," in *ICDE*. IEEE, 2025, pp. 99–113. [Online]. Available: https://doi.org/10.1109/ICDE65448.2025.00015

[70] Y. Zhong, T. N. Chan, L. H. U, D. Wu, W. Tu, R. Wang, and J. Z. Huang, "A fast and accurate block compression solution for spatiotemporal kernel density visualization," in *SIGKDD*. ACM, 2025, pp. 4098–4109. [Online]. Available: https://doi.org/10.1145/3711896.3736821

[71] "Seattle geodata," https://data.seattle.gov/Transportation/SDOT-GISDatasets/jyjy-n3ap.

[72] "Atlanta crime data (2009 - 2020)," https://experience.arcgis.com/experience/76b68b923a094fb5a2d193ace5ccb975/.

[73] "Los angeles trip data," https://bikeshare.metro.net/about/data/.

[74] "NYC open data," https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95.

[75] E. Delmelle, C. Dony, I. Casas, M. Jia, and W. Tang, "Visualizing the impact of space-time uncertainties on dengue fever patterns," *International Journal of Geographical Information Science*, vol. 28, no. 5, pp. 1107–1127, 2014. [Online]. Available: https://doi.org/10.1080/13658816.2013.871285.