GeoAI for Emerging Spatial Datasets

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Yan Li

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Shashi Shekhar

May, 2022

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. Shashi Shekhar, for his guidance, support, and patience throughout my Ph.D. Prof. Shekhar introduced me into the world of spatial computing, and encouraged me to investigate interesting, important, and challenging problems. He not only helped me to develop my technical skills, but also helped me to understand how to do research. I would also like to thank all the professors who helped me over the years and those who served on my committee: Prof. Jaideep Srivastava, Prof. William Northrop, Prof. Yao-Yi Chiang, and Prof. Jesse Berman. Thank you for showing me the correct direction to shape my research.

I am grateful for my friends in Prof. Shekhar's Spatial Computing research group for all the great insights they provided. Particularly, I thank Emre and Zhe for mentoring me to participate in research in my first year of the Ph.D. I would also thank Yiqun, Majid, and Mingzhou for their great help in my research. I thank Kim Koffolt for reading drafts of my papers and sharpening my writing. She taught me how to write good research papers in English.

Lastly, I would like to thank my family. I thank my girl friend Xiaochen for her love and support. Together, we make it through trial and error, happiness and sorrow in my Ph.D. None of this would happen without her. I also want ot express my gratitude to my parents. They are the best parents in the world, and I learn how to be a man from them.

Dedication

To my girl friend, Xiaochen Liu. To my parents, Chengbin Li and Qinxia Yan. To all my friends.

Abstract

Geospatial artificial intelligence (GeoAI) is the generalization of conventional artificial intelligence (AI) to meet the challenges posed by spatial data. Spatial data, i.e., data annotated with spatial information such as locations and shapes, has been growing available over the last decade and transformed lives by providing novel ways of observing the world, knowing places and the relations between them. For example, large amount of onboard diagnostics data from vehicles becomes available with the popularity of telematics devices equipped with GPS chips and makes monitoring vehicles' real-world performance possible, which is valuable for domains such as vehicle mechanics, transportation science, and city planning. In many other domains such as smart city and public health, spatial data becomes critical as well. For example, during the Covid-19 pandemic period, mobile tracking data from devices with GPS chips has been used as an important way of contact tracing and traveling pattern surveying. A McKinsey Digital report estimates that personal spatial data could help save consumers about $600 billion by 2020.

Recent years have witnessed significant advances in AI in both academia and industry. Its fast development is powered by big data and high-performance computing platforms that support the development, training, and deployment of AI methods with reasonable cost.

Even though spatial data are critical, valuable, and collected in a large scale, and AI techniques have been applied to many problems such as computer vision and natural language processing successfully, spatial data pose great challenges to conventional AI techniques. The first challenge is the gap between AI techniques and domain knowledge. Conventional AI techniques rarely consider domain knowledge (e.g., physics laws and epidemiology models), making their results hard to interpret and susceptible to violate domain constraints even with large volumes of data. On the other hand, domain knowledge by itself is insufficient due to its reliance on simplifying assumptions that may not approximate the complex real-world scenarios well. The other challenges are caused by the properties of spatial data, namely, spatial autocorrelation, spatial heterogeneity, and spatial continuity. Spatial autocorrelation describes the fact that the data samples (e.g., temperature, precipitation) at different spatial locations are correlated with each

other and are affected by their geographical neighbors, which violates the common i.i.d. (i.e., independent and identical distribution) assumption underlying many machine learning models. Spatial heterogeneity refers to the fact that the data samples at different spatial locations are different from each other, so there may not be universal models that are applicable globally. Spatial continuity refers to the fact that the conflict between the continuity of the geographic space and the discrete representation of spatial data.

This thesis investigates novel and societally important GeoAI techniques for emerging spatial datasets such as multi-attributed trajectories and categorical point sets. Multiple novel approaches are proposed to address challenges posed by the datasets on conventional AI techniques. Specifically, a Quad-Grid Filter & Refine algorithm is introduced to detect local spatial colocation patterns, which consider the spatial heterogeneity property of colocation patterns. The algorithm can detect colocation patterns that may not be prevalent globally but are prevalent in local regions, and it is much more computationally efficient than the baseline algorithm. Second, the thesis investigate the problem of discovering contrasting spatial colocation patterns that have different prevalence in two groups of spatial datasets. It leverages the domain knowledge that neighborhood relationships between categorical spatial objects may convey important information, and introduces a filter & refine algorithm using the anti-monotone property of a proposed metric to measure the prevalence difference of any colocation patterns in the two groups. Third, the thesis discusses a point-set classification method for multiplexed pathology images. Inspired by the domain assumption that the spatial configuration of cells may vary under different health conditions, this thesis introduces a neural network architecture to capture the spatial configurations of categorical point sets through modeling pairwise relationships. Last, the thesis introduces a physics-guided K-means algorithms to estimate the energy consumption for a vehicle to travel along a path, which is a combination of physics laws followed by vehicle energy consumption and a machine learning model. The thesis also proposes a path-centric path selection algorithm using the proposed energy consumption estimation model considering the spatial autocorrelation property of the data.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Geospatial artificial intelligence (GeoAI) is the generalization of conventional artificial intelligence (AI) to meet the challenges posed by spatial data, i.e., data annotated with spatial information such as locations and shapes.

1.1 Spatial Data

Over the last decade, there has been a significant growth in the availability of spatial data, which transforms lives by providing novel ways of observing the world, knowing places and the relations between them [1].

The most traditional way of acquiring spatial data is surveying, which has been conducted since the beginning of recorded history [2]. The function of surveying includes determining and measuring spatial objects, assembling information related to spatial objects, and using the information for planning. Due to its societal importance and high cost to conduct, surveying was mostly conducted by government agencies, which limits the public access to spatial data.

Since remote sensing techniques emerged in 1960s, they have become another important source of spatial data. With the development of remote sensing platforms (e.g., satellites, airplanes, and UAVs) and sensors (e.g., multispectral scanner, LiDAR scanner, RADAR sensor), remote sensing techniques with higher spatial, spectral, and temporal resolution are applied in the last decade [3]. Spatial resolution of remote sensing imagery refers to the size of the area covered of a pixel in an image, so images

with high spatial resolution give us an opportunities to detect fine-grained spatial objects such as buildings and trees. Spectral resolution refers to the number of spectrum ranges that can be recorded by sensors. Different spectrum ranges convey different information about spatial objects and are complementary with each other [4]. For example, images of visible spectrum are easy to read by humans, while microwaves used by RADAR have excellent penetration capacity that enable it to work in various weather conditions and to detect targets under the Earth's surface [5], and LiDAR point cloud contains much information about the shape of spatial objects [6]. Temporal resolution means the frequency of an area be covered by spatial data. High temporal resolution spatial data can be used to detect spatial objects more accurately using their temporal fingerprints, and can also be used to monitor changes of objects.

The last decade also witnesses the integration of portable computing devices (e.g., smart phones, watches) and GPS chips, which has boosted spatial data generation through crowdsourcing. For example, OpenStreetMap is a crowdsourced map built by volunteers around the world, which provides base maps to a large number of spatial data science research (e.g., [7]). Applications that users can use to check in with their locations such as Yelp and Twitter provide spatial data with rich non-spatial attributes and attract considerable research interest [8]. The emerging crowdsourced spatial data are mostly in two data types, namely, point sets and multi-attributed trajectories. A point set refers to a collection of points in a 2/3-dimensional geographic space. Each point is associated with multiple non-spatial attributes and represents a spatial object or an event. For example, a point set can represent a collection of points of interest (POIs) in a city, and each point represents a POI (e.g., a restaurant, a movie theater) and is associated with attributes such as business hours, grand names, etc [9]. An application of point-set spatial data in public health is that researchers can use point sets that represent the cases of certain diseases to detect clusters of the cases so as to discover potential causes to the diseases [10]. A multi-attributed trajectory is a sequence of multi-attributed points in a 2/3-dimensional geographic space that records the status of a moving object along a journey. Onboard diagnostics (OBD) data from vehicles is an example of multi-attributed trajectories, which record dozens of engine measurements (e.g., state of charge, RPM, etc.) and vehicle locations at each timestamp during their trips [11]. Before OBD data became available because of the popularity of onboard telematics devices, transportation

scientists can only evaluated vehicles in controlled laboratory experiments and test track studies, which do not adequately predict the performance of vehicles during real-world driving [12]. In public health, crowdsourced spatial data attract growing attention as well. For example, during the Covid-19 pandemic period, mobile tracking data from devices with GPS chips has been widely used as a way of contact tracing and traveling pattern surveying [13].

1.2 GeoAI

The large-scale availability of spatial data boost recent progress in geospatial artificial intelligence (GeoAI) [14]. For instance, remote sensing images of high spatial resolution and deep learning computer vision methods fuel progress in fast and accurate spatial object detection ([15, 16]).

Depending on the goal to accomplish, there are four primary types of GeoAI tasks, namely, descriptive, diagnostic, predictive, and prescriptive tasks. Descriptive GeoAI tasks focus on revealing valuable insight from spatial data, which can be in the form of data visualizations like graphs, charts, reports, and dashboards. For example, in public safety discovering regions where the density of crimes is statistically higher than other places helps allocate the police more efficiently [17, 18]. In transportation science, detecting road segments along which vehicles spend significantly more energy than that on any other road segments helps researchers to design more energy efficient roads and vehicles [19]. In material science, discovering the clusters of particles with unusual rotations helps to analyze the energy transmission across a piece of material [20]. Diagnostic GeoAI tasks look for cause and effect to illustrate why something happened. They compare the occurrences of events or the existence of objects to determine the causes to the occurrences or existence. Back to the transportation science example, now that we are aware that the energy consumption of vehicles is significantly higher along certain roads, a diagnostic task is to identify the potential causes to it. Additional techniques are needed to discover the correlation between the high energy consumption and other factors such as terrain and snow accumulated on roads [9]. Predictive tasks tell what is likely to happen according to the key trends and patterns in historical data. Back to our transportation science example, according to the features of a road

segment (e.g., length, elevation change, road type), we may want to predict the energy consumption of a vehicle traveling along it using historical energy consumption data for other roads and vehicles [12]. The last group of tasks is prescriptive tasks, which focus on finding optimal solutions according to historical data and predictions. Back to our transportation science example, now that we know the predicted energy consumption of a vehicle traveling along all road segments, a sample prescriptive task is to find an energy-efficient path between two given places where the traveling energy consumption is the lowest when compared with any other candidate paths between the two places [21].

1.3 Illustrative Application Domain I: Smart Transportation

GeoAI technologies have transformed how people travel with an ever-growing set of tools such as Google Maps, Uber, etc. These tools change the way people understanding the world, knowing and communicating relations to places, and navigating through these places. Smart transportation refers to a transportation system that apply a variety of technologies (e.g., smart phones, traffic cameras, air quality sensors) to monitor, evaluate, and manage transportation systems to enhance efficiency and safety. With the great variety of spatial data (e.g., onboard diagnostics data from vehicles) being collected at both larger scales and higher resolution, there exists a lot of opportunities for GeoAI to provide timely solutions to critical transportation problems.

One major problem faced by transportation is sustainability, which has attracted growing more attention. Sustainable transportation aims to reduce the environmental impact of vehicles by improving energy efficiency and reducing toxic emissions. Transportation accounts for the vast majority of US petroleum consumption as well as over a third of greenhouse gases and over a hundred thousand U.S. deaths annually via air pollution [22]. Large amount of effort has been made to reduce energy consumption, such as the use of regenerative braking and auto stop-start engines, as well as the innovation of electric cars. However, the expected energy use continues to climb. The U.S. Department of Energy predicts world energy consumption for transportation will rise 28% between 2015 and 2040 [23]. Controlled laboratory experiments and test track studies that are commonly used in transportation science do not adequately predict

emissions and energy-consumption during real-world driving. This fact is illustrated by the Volkswagen emissions scandal, fines levied on other manufacturers and the subsequent move away from diesel and gasoline energy towards electrified vehicles in many countries. Thanks to the development of telematics devices with GPS, which collect large volumes of onboard diagnostics data, monitoring the vehicles under real-world driving conditions becomes possible. Preliminary evidence for the potential of reducing energy consumption and greenhouse gas emission through GeoAI includes the experience of UPS, which saves around a million gallon of fuel every year by preferring routes that avoid left turns. As this thesis will show, GeoAI has the potential to improve the sustainability of transportation by suggesting energy-efficient paths according to the historical onboard diagnostics data, vehicle model information, and other auxiliary spatial data (e.g., road maps, weather) [21].

1.4 Illustrative Application Domain II: Spatial-informed Pathology

GeoAI can also help in the pathology field. Pathology is the study of the causes and effects of disease or injury. In pathology, the standard procedure to diagnose many diseases, including cancers, are biopsies. In this procedure, a tissue sample is removed from the body, chemically treated, sliced into thin sections, and placed on a glass slide, and stained with specific chemicals to enhance contrast for visual inspection [24]. A pathologist then performs a macroscopic examination of the specimen and describes various features such as type of cells present, their distribution, and other important diagnostic features.

The developments in whole slide digital imaging and antigen-based staining technology enable identification of up to more than 30 markers on each cell based on the cell's surface chemistry with high throughput [25, 26]. These novel technologies have played an important role in the era of cancer immunotherapeutic treatment regimens [27, 28]. Immunotherapy involves the treatment of diseases by inducing, enhancing, or suppressing an immune response in the patient. This treatment regimen has been gaining increasing attention due to its potential in the treatment of cancers which are non-responsive to conventional methods such as radiotherapy and chemotherapy [29, 30]. As this treatment

regimen utilizes the immunoregulatory cells of the patient in eliminating tumorous cells, there is a growing interest in understanding the interplay between various cells in a spatially informed manner in the tumor microenvironment [31, 32]. For example, for tumor infiltrating lymphocytes (TILs) to be able to induce cell death, these cells must have direct or proximal contact with tumor cells. Thus, the distance between tumor and immune cells is an important indicator for determining disease progression and treatment effect, and emerging research in this area has begun to highlight the importance of spatial organization among cell phenotypes for cancer diagnosis and prognosis [32]. The development and adoption of spatially informed methods both for tumor and disease micro-environment quantification generally would help in developing optimal treatment plans tailored to each patient. Additionally, it would be prudent to leverage the power of algorithmic intelligence in the pathology domain, as it can provide insights which cannot be captured visually by a pathologist.

1.5 Challenges

While spatial data are critical, valuable and collected at massive scales, they pose great challenges to conventional AI techniques and platforms when applied to important societal problems.

1.5.1 Domain Knowledge Gap

The conventional AI techniques often do not consider domain knowledge (e.g., laws of physics, epidemiology models), making their results hard-to-interpret and susceptible to domain constraint violations even with large volumes of data. On the other hand, domain knowledge by itself is also insufficient due to its reliance on simplifying assumptions that may not be well-suited for complex real-world scenarios. This calls for a more holistic view to solve real-world problems by leveraging both data-driven techniques and scientific domain understanding [33].

1.5.2 Properties of Spatial Data

Other challenges come from the properties of spatial data, such as spatial dependence and spatial heterogeneity. Tobler's first law of geography, "everything is related to

everything else, but near things are more related than distant things" [34], describes the spatial dependence that ubiquitously exists in the phenomena on earth. For example, people living in the same neighborhood tend to share similar characteristics, income, and education level. In spatial statistics, spatial dependence is called the spatial autocorrelation effect. Ignoring autocorrelation and assuming an identical and independent distribution (i.i.d.) of data when analyzing spatial data may produce hypotheses or models that are inaccurate [35]. For example, applying non-spatial machine learning methods (e.g., random forest) on land cover segmentation using remote sensing imagery may result in salt and pepper noise [36]. Spatial dependence exists not only at close locations, but also distant locations. One example of long-range spatial dependence is El Nino and La Nina effects in the climate system. Spatial heterogeneity refers to the fact that spatial data do not follow an identical distribution throughout the entire earth [9, 37]. For example, the appearance information of European and Spiny Toads are visually similar, but they are located in different geographical regions and belong to different species [38], which makes "one-size-fits-all" models using appearance information only hardly applicable. Furthermore, while conventional AI techniques need discrete input data, for example, transactions in association rule mining, spatial datasets are embedded in continuous space, which makes the non-spatial techniques inapplicable. The discretization of space may introduce problems such as the modifiable area unit problem (MAUP) or the multi-scale effect, since the results of spatial analysis depend on the choice of discretization methods and spatial scale. Figure 1.1(a) shows three input spatial feature types A, B, and C and the nearby relationship in between. Depending on the choice of discretization methods as shown in Figure 1.1(b) and (c), the correlation coefficients of the pairs (A,B) and (B,C) are -1 and 1 respectively. Gerrymandering, which is a practice intended to establish a political advantage for a particular party or group by manipulating district boundaries, is a form of the MAUP, and is attracting growing attention in recent years [39, 40].

1.6 Thesis Contributions

This thesis investigates GeoAI techniques for emerging spatial datasets. The contributions are summarized in a taxonomy shown in Table 1.1 which describes the input spatial

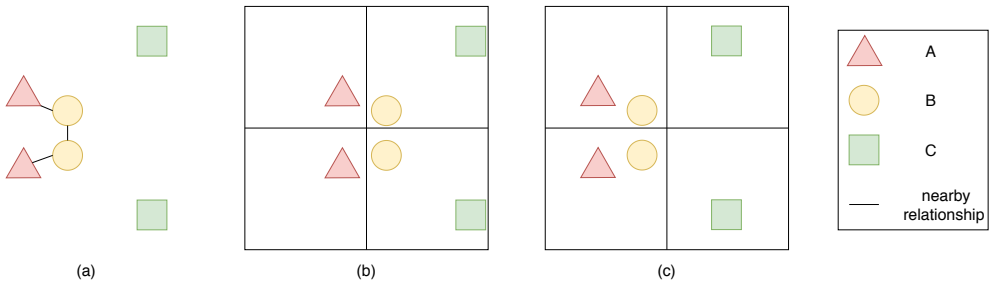


Figure 1.1: Example of the modifiable area unit problem.

Table 1.1: Thesis contribution taxonomy.

		GeoAI tasks			
		Descriptive	Diagnostic	Predictive	Prescriptive
Data type	Point set	Local colocation (chapter 2)	Contrasting colocation (chapter 3)	Point set classification (chapter 4)	
	Multi-attributed trajectory				Eco-routing (chapter 5)
	...				

data types and the GeoAI tasks they are used to solve. The input spatial data are in two data types, namely, point sets and multi-attributed trajectories. Spatial data in the format of point sets can be used to represent the existence of spatial objects or the occurrence of events in 2/3-dimensional geographic space. For example, a point set can be used to represent the locations of cells on a stained tissue sample in the pathology field. Spatial data in the format of multi-attributed trajectories can be used to represent the status of moving objects along journeys. For example, onboard diagnostics data from vehicles are in the form of multi-attributed trajectories. Each point in a trajectory records the instantaneous status of a vehicle. The content of each chapter is briefly introduced below.

- Chapter 2 discusses a novel spatial colocation pattern detection approach, local

colocation pattern detection (LCPD), to find pairs of spatial colocation patterns and regions such that the spatial colocation patterns are prevalent in their paired regions [9]. The approach addresses the limitations of traditional spatial colocation pattern detection approaches which ignore spatial heterogeneity but assume that the instances of spatial colocation patterns are evenly distributed in the study area. LCPD has many societal application fields such as forestry study where the distribution of tree species is heavily affected by the environment. LCPD proposes multiple novel algorithms, namely quadruplet and QGFR algorithms, to successfully handle the computational challenges caused by large number of candidate regions, non-monotonicity of the test statistic. To balance between computational tractability and richness of the pattern, LCPD uses minimum orthogonal bounding rectangles (MOBRs) of spatial colocation pattern instances as approximations of arbitrary-shape regions. Under this assumption, LCPD enumerates all MOBRs of colocation pattern instances in the given study area and guarantees correctness and completeness of the solution. Theoretical and experimental analyses show that the proposed algorithms yield substantial computational savings compared to baseline approaches. Case studies demonstrate that LCPD can find local colocation patterns that are not prevalent globally in the study area.

- Chapter 3 investigates the problem of contrast spatial colocation pattern detection (CSCPD) in two groups of spatial datasets whose prevalence is substantially different in the two groups. The approach is important for a variety of application domains such as pathology where different spatial colocation patterns of tumor cells and immune cells may indicate different stages of diseases. CSCPD introduces a metric to describe the difference between the prevalence of spatial colocation patterns in two groups of spatial datasets based on a commonly-used prevalence metric of spatial colocation patterns, and then proposes a filter & refine algorithm utilizing the anti-monotone property of the proposed metric without affecting the completeness and correctness of the results. Extensive experiments indicate that the proposed algorithm yields substantial computational time savings. A case study on a real-world dataset derived from multiplexed immuno-fluorescence images shows that the proposed method is capable of finding patterns that are ignored by the related work and has the potential to advance scientific discovery.

- Chapter 4 investigates a point-set classification method for multiplexed pathology images, which aims to distinguish between the spatial configurations of cells within multiplexed immuno-fluorescence (mIF) images of different diseases [41]. This problem is important because it provides a novel way for pathologists to diagnose diseases according to the interactions between cells. This problem is challenging because crucial spatial relationships are implicit in point sets and the non-uniform distribution of points makes the relationships complex. Manual morphologic or cell-count based methods, the conventional clinical approach for studying spatial patterns within mIF images, is limited by inter-observer variability. In this chapter, a new deep neural network architecture, namely spatial-relationship aware neural networks, is proposed. Experimental results with a University of Michigan mIF dataset show that the proposed method significantly outperforms the competing deep neural network methods, by 80%, reaching 95% accuracy.

- Chapter 5 investigate the problem of selecting energy-efficient path using historical onboard diagnostics data [12, 11, 21]. It is an societally critical problem since the sustainability and prosperity of cities benefit from reducing energy consumption of transportation. The problem is challenging because the expected energy consumption of a vehicle depends on its physical parameters and follows physics laws, and there exists autocorrelation between the energy consumption on road segments. This chapter introduces a physics-guided K-means algorithm to estimate the energy consumption of vehicles on paths, and a maximal-frequented-path-graph shortest-path algorithm, and an informed algorithm using an admissible heuristic. Theoretical and experimental evaluation shows that the proposed algorithms yield substantial computational time savings, and that they select paths that are more energy-efficient than the paths selected by the state-of-the-art methods.

- Chapter 6 summarizes the thesis findings and gives an overview of related directions and topics for research in the future.

Chapter 2

Local Co-location Pattern Detection

2.1 Introduction

Given instances of different spatial features (e.g., mall, hospital) and a spatial relation, the problem of local co-location pattern detection (LCPD) pairs co-location patterns and localities such that instances of the features in a co-location pattern tend to be related to each other inside the paired locality. Intuitively, if a co-location pattern is infrequent relative to all input instances, it may be neglected in the entire dataset, but more easily found in a subset of the dataset around its spatial footprint. The uneven distribution of spatial features in the space, i.e., spatial heterogeneity, is common, so the local existence of co-location patterns in an area is not unusual. For example, high NOx emissions from buses may occur with certain engine events only around the bus depot where the route starts, since the engines have not warmed enough to perform efficiently. Other examples include high NOx emission and elevation change in rural areas as illustrated in the Volkswagen emissions scandal [42], and assault crimes and drunk driving events near bars [43]. Because of its societal importance, LCPD has attracted growing attention recently.

In this chapter, we will focus on detecting local co-location patterns with the locality defined using minimum orthogonal bounding rectangles (MOBRs). An MOBR is a rectangle with sides parallel to the coordinate system. It is widely used as an

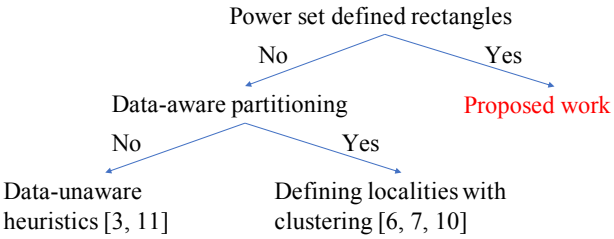


Figure 2.1: The related work.

approximation of complex shapes by minimally enclosing them [44]. However, the enumeration of MOBRs is computationally challenging. Given a set of spatial objects in a 2-dimensional space, the number of the set's subsets is exponentially related to its cardinality. Each of the subsets has an MOBR, so the number of MOBRs is also exponentially related to the number of the input objects. Moreover, the relationship between the participation index, a widely adopted metric for co-location patterns [45], in any pair of localities cannot be determined without considering the distribution of spatial objects within them.

2.1.1 Related Work and Limitations

In order to solve the LCPD problem, many methods have been proposed, which can be generalized into two steps. The first step is partitioning the study area into potential localities based on certain heuristics, which is followed by checking the eligibility of the localities. Based on whether the heuristics are data-aware, these methods belong to two classes (the right branch in Figure 2.1).

A good example using data-unaware heuristics is [46] in which Celik et al. use a QuadTree structure to divide the study area into localities, but it requires sophisticated domain knowledge to predefine localities. In another example, a grid is used to divide the study area into cells, and arbitrary subgraphs of the cells' neighbor graph are regarded as localities [47]. Both approaches share the same limitation with others using data-unaware heuristics, that is, the partitioning scheme employed is independent of the spatial distribution of the data, which may break up potential localities [43].

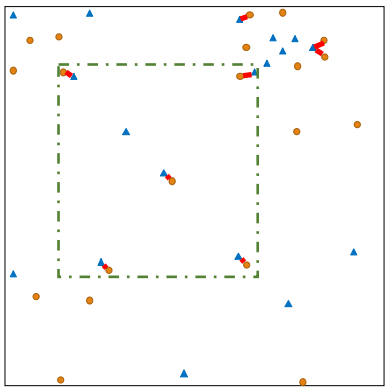
The other class of methods using data-aware heuristics defines localities with clusters of spatial objects or co-location instances. In [48], localities grow from initial localities

with high objects concentration. Mohan et al. define localities as areas delineated by neighbor graphs of spatial objects [43]. Deng et al. explore footprints of co-location instance clusters with an adaptive density threshold as localities [49]. These methods are not complete because localities without object or co-location instance concentrations may be eligible as well.

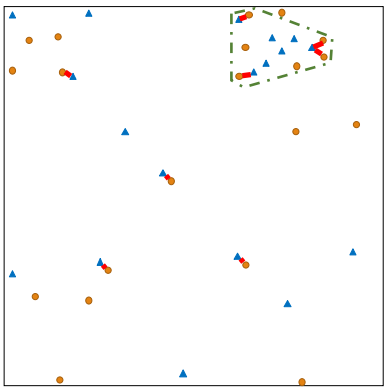
Our proposed work, on the other hand, detects local co-location patterns in all rectangular localities with sides parallel to the coordinate system, so the method will enumerate the MOBRs determined by all subsets of co-location instances (the elements in co-location instances' power set). Consider the dataset shown in Figure 2.3 as an example. If the participation index threshold is set as 0.6, the co-location pattern $\{f_A, f_B\}$ is not a eligible pattern globally through the data, because its participation index is $\frac{7}{18}$. However, our proposed work will find a prevalence locality for the pattern (green dash rectangles in Figure 2.2a), where the participation index is $\frac{5}{6}$. Contrarily, The participation index in the locality determined by the cluster of co-location instances shown in Figure 2.2b is $\frac{3}{7}$, while Figure 2.2c and 2.2d present the localities with the highest possible participation index if the study area is partitioned using the Quadtree and grid in them, where the participation index is $\frac{3}{7}$ in both cases. None of the currently available results in eligible patterns, so it is obvious that the proposed work will detect more complete results than the relate work.

2.1.2 Contributions

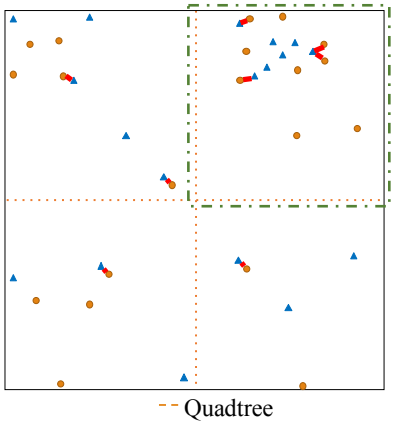
To detect local co-location patterns in all rectangular localities with sides parallel to the coordinate system, we first formally define the LCPD problem. Then, we present a Quadruplet & Grid Filter-Refine algorithm that leverages an MOBR enumeration lemma, and a novel upper bound on the participation index. The experimental evaluation shows that the proposed algorithm reduces the computation cost substantially. One case studies on North American Atlas-Hydrography and U.S. Major City Datasets was conducted to discover local co-location patterns which would be missed if the entire dataset was analyzed or methods proposed by the related work were applied.



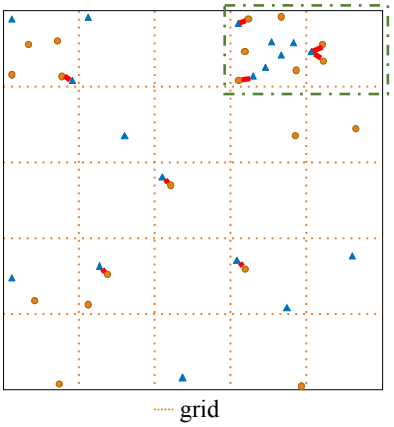
(a) Proposed work.



(b) Data-aware heuristic using clustering.



(c) Data-unaware heuristic using Quadtree.



(d) Data-unaware heuristic using a grid.

Figure 2.2: Comparison between related work. (Better in color.)

2.2 Basic Concepts and Problem Statement

2.2.1 Basic Concepts

Huang et al. define the input, output and the interest measures for detecting co-location patterns globally through data in [45].

Each spatial **object**, composed of a boolean **feature** (e.g., mall, hospital) and a spatial location, can be related to others through a spatial **relation** (e.g., neighborhood). A **co-location pattern** is a set of features. An instance of a co-location pattern is a set of objects of every distinct feature in the pattern which can form a clique given the input relation. In the dataset shown in Figure 2.3, there are 20 objects of feature f_A (circle) and 18 objects of feature f_B (triangle), and the related objects are linked. Only one co-location pattern, $\{f_A, f_B\}$, exists, and it has 8 instances.

The **participation ratio** of a feature f_i in a co-location pattern C, $pr(C, f_i)$, is the fraction of objects of the feature participating in instances of the pattern. The **participation index** of the pattern, $pi(C)$, is the minimal participation ratio of the features in the pattern. In Figure 2.3, for the co-location pattern $C = \{f_A, f_B\}$, $pr(C, f_A) = \frac{8}{20}$ and $pr(C, f_B) = \frac{7}{18}$, so $pi(C) = \frac{7}{18}$.

By extending these concepts, we introduce the following ones for the LCPD problem.

The **study area** is defined as the minimum orthogonal bounding rectangle (MOBR) of all input objects, whose subsets are **localities**. A **local co-location pattern** is a pair of a co-location pattern (C) and a locality (r), in the form of $< C, r >$. Its instances and interest measure are the corresponding values of its co-location pattern in its locality. A locality where objects of features in a co-location pattern tend to be related to each other (determined by a participation index threshold) is called the pattern's prevalence locality.

In Figure 2.3, for a local co-location pattern $C_r =< \{f_A, f_B\}, r >$, there are 5 instances, while $pr(C_r, f_A) = \frac{5}{5}$, $pr(C_r, f_B) = \frac{5}{6}$, and $pi(C_r) = \frac{5}{6}$. If the participation index threshold is 0.5, r is a prevalence locality of the pattern $\{f_A, f_B\}$.

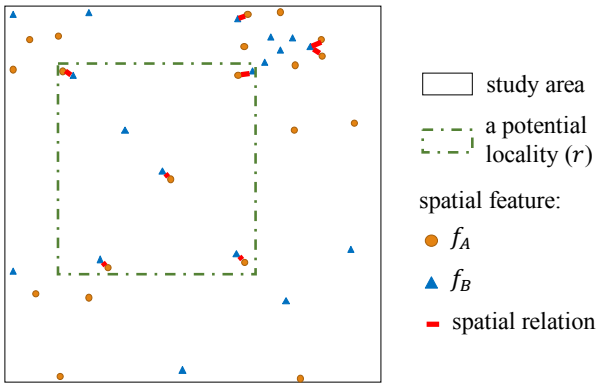


Figure 2.3: A local co-location pattern $< \{f_A, f_B\}, r >$.

2.3 Approach

We first introduce a baseline algorithm for the LCPD problem. Then, we present two refinements: a Quadruplet (Quad) algorithm as well as a Quadruplet & Grid Filter-Refine (QGFR) algorithm, to reduce the computational cost without impairing correctness and completeness.

The pseudo-code of the general algorithm framework is shown in Algorithm 1. In this framework, all possible co-location patterns of the features associated with the input objects are enumerated in line 2-11. The instances of each co-location pattern are generated as the input of an MOBR-generating function MOBRGenerator (line 4), and the MOBRs obtained from this function are enumerated to detect the prevalance ones (line 4-10). Consider the dataset in Figure 2.3 as an example. In this case, F has two elements: f_A and f_B, so there is only one possible co-location pattern, $\{f_A, f_B\}$, whose 7 instances are saved in CI (line 3). The locality r is one of the MOBRs to be enumerated. There are 5 instances within it, and the participation index is $\frac{5}{6}$. Both metrics will be compared with the thresholds to determine whether $< \{f_A, f_B\}, r >$ is an eligible result.

In this study, we focus on reducing the number of MOBRs enumerated for each co-location pattern (i.e., improving function MOBRGenerator($\cdot$)), but adopt Apriori-like algorithms to reduce the number of possible co-location patterns [45, 49], and the state-of-the-art algorithms to generate co-location instances [45, 50].

Algorithm 1 General algorithm framework

Require:
Obj: A set of objects;
R: A spatial relation over objects in Obj;
θ: Participation index threshold;
γ: Co-location instance number threshold.

Ensure: Local co-location patterns with participation index $\geq \theta$ and the number of instances $\geq \gamma$.

```
1: F ← all spatial features in Obj;
2: for all possible patterns C of F do
3:     CI ← co-location instances of C;
4:     for all mobr ∈ MOBRGenerator(CI) do
5:         p ← the participation index of C in mobr;
6:         n ← the number of C's instances in mobr;
7:         if p ≥ θ and n ≥ γ then
8:             Add < cp, mobr > to the result.
9:         end if
10:    end for
11: end for
```

2.3.1 Baseline Algorithm

As already mentioned, we focus on localities defined as the MOBRs of subsets of co-location instances. In the function MOBRGenerator($\cdot$) of the baseline algorithm, we will enumerate all arbitrary subsets of the input co-location instances, and generate an MOBR for each of them. If each co-location pattern has n_{ci} instances on average, there will be $2^{n_{ci}}$ subsets, resulting in $2^{n_{ci}}$ MOBRs. Thus, the computational complexity of this baseline algorithm is $O(k2^{n_{ci}})$, where k is the number of possible co-location patterns.

2.3.2 Quad-Element Algorithm

Our first improvement is based on an MOBR enumeration lemma:

Lemma 2.3.1. *Given a set s of n points in a two-dimensional plane, the set of MOBRs for arbitrary subsets of s is the same as the set of MOBRs for arbitrary subsets with cardinality ≤ 4 of s.*

Proof. Assume that there exists an MOBR for a subset (*sub*) with cardinality > 4 that

is not an MOBR for a subset with cardinality ≤ 4.

Let $x_{min}, x_{max}, y_{min}, y_{max}$ denote the minimum and maximum of the x, y coordinates of the points in *sub*. There must exist points a, b, c, and d (which may be the same) in *sub* such that $x_a = x_{min}, x_b = x_{max}, y_c = y_{min}, y_d = y_{max}$. Thus, the MOBR for *sub* is the same as that for $\{a, b, c, d\}$, which is a subset of s with cardinality ≤ 4, resulting in a contradiction with the assumption. □

Lemma 2.3.1 indicates that the enumeration cost of a co-location pattern's MOBRs can be reduced from 2^n to n^4 without affecting completeness. By changing the function MOBRGenerator$(\cdot)$ to generate the MOBRs of subsets with cardinality ≤ 4 of CI we can get the Quadruplet (Quad) algorithm with computational complexity of $O(kn_{ci}^4)$.

2.3.3 Quadruplet & Grid Filter-Refine Algorithm

Our definition of localities determines that a small displacement of any co-location instance that defines a locality's boundary will create a new locality, so there are lots of localities overlapping each other. If we can classify them into groups according to the areas they share, and apply a filter on each group instead of on individuals, the number of localities to be enumerated can be reduced further. Based on this idea, we proposed the second improvement: the Quadruplet & Grid Filter-Refine (QGFR) Algorithm.

The pseudo-code of the function MOBRGenerator$(\cdot)$ in the QGFR algorithm is shown in Algorithm 2. Because a grid-based filter is applied, three new parameters are added, namely, a threshold of the participation index, a threshold of the number of co-location instances, and the cell size of the grid covering the entire study area. The first step of the function is saving the active cells of the input co-location pattern C (i.e., the cells overlapping C's instances) in AC (line 2). A cell overlapping a co-location instance means that the intersection of the cell and the MOBR of this instance is nonempty. For example, Cells 1, 2, and 3 in Figure 2.4 are active cells of the pattern $\{f_A, f_B\}$. After getting the active cells, we will use their MOBRs (cMOBR) as an approximation of the MOBRs of C's instances (iMOBR). The cells in a cMOBR are classified into two parts. The cells adjacent to the cMOBR's boundary are named as *bounding* cells, while the others are the *bounded* cells. In Figure 2.4, a cMOBR is delineated by a red solid rectangle, while its bounding and bounded cells are filled with a hash pattern and a

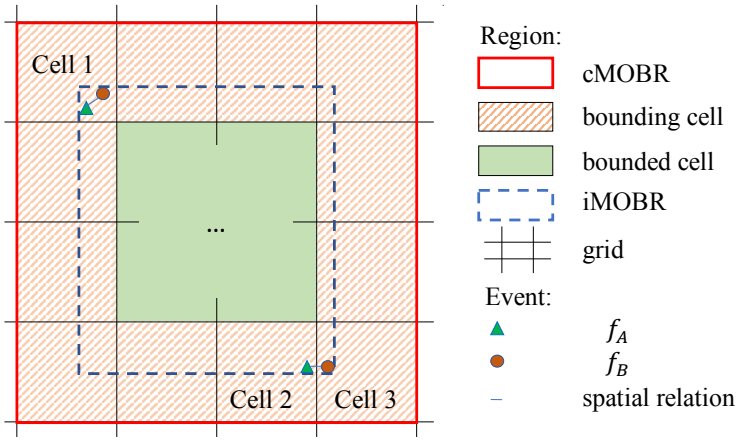


Figure 2.4: Grid cells and MOBRs (better in color).

solid color respectively. The boundary of each iMOBR has the following property:

Lemma 2.3.2. *The boundary of any iMOBR must be within the bounding cells of one and only one cMOBR.*

The proof of this lemma is straightforward. If the boundary of an iMOBR is not within the bounding cells of a cMOBR, at least one of its four edges does not pass active cells, which is impossible. If two cMOBRs share the same bounding cells containing an iMOBR's boundary, they must be the same. Therefore, we define that an iMOBR is in a cMOBR if its boundary is within the bounding cells of the cMOBR. For example, an iMOBR delineated by a dash rectangle in Figure 2.4 is in the plotted cMOBR. Because each iMOBR is in a unique cMOBR, by enumerating the iMOBRs in each cMOBR, we can enumerate all iMOBRs just once. In the pseudo-code, we enumerate all cMOBRs using Lemma 2.3.1 (line 3-10).

To eliminate the cMOBRs in which no iMOBR is eligible, we introduce an upper bound (MaxPI bound), $\eta(< C, \text{cMOBR} >)$, for the participation index of a local co-location pattern composed of a co-location pattern C and any iMOBR in a cMOBR of C. The MaxPI bound is based on an upper bound for the participation ratio, which can be stated as:

Lemma 2.3.3. *The upper bound, $\zeta(< C, \text{cMOBR} >, f)$, for the participation ratio of a feature f in a local co-location pattern composed of a pattern C and any iMOBR in a*

cMOBR of C is

$$\zeta(< C, cMOBR >, f) = \frac{po(C, f, \text{cMOBR})}{o(f, \text{bounded}) + po(C, f, \text{bounding})}$$

$\forall$ *iMOBR in cMOBR.*

Table 2.1 describes the notation used in the above formula.

Table 2.1: Symbols used in Lemma 2.3.3.

Number of objects of f in a locality r		
Participating in C	Not participating in C	All
$po(C, f, r)$	$npo(C, f, r)$	$o(f, r)$

where r can take values of "all cells" (cMOBR), "bounding cells" (bounding), or "bounded cells" (bounded) of the cMOBR, or the "actual iMOBR" (iMOBR), or the "intersection of iMOBR and bounding cells" (extra). The proof is as follows:

Proof.

$$\begin{aligned} pr(< C, iMOBR >, f) &= \frac{po(f, C, \text{iMOBR})}{o(f, \text{iMOBR})} = \frac{po(f, C, \text{bounded}) + po(f, C, \text{extra})}{o(f, \text{bounded}) + o(f, \text{extra})} \\ &= \frac{po(f, C, \text{bounded}) + po(f, C, \text{extra})}{o(f, \text{bounded}) + po(f, C, \text{extra}) + npo(f, C, \text{extra})}. \end{aligned}$$

Because $npo(f, C, \text{extra}) \geq 0$,

$$pr(< C, iMOBR >, f) \leq \frac{po(f, C, \text{bounded}) + po(f, C, \text{extra})}{o(f, \text{bounded}) + po(f, C, \text{extra})}.$$

Because extra $\in$ bounding, $0 \leq po(f, C, \text{extra}) \leq po(f, C, \text{bounding})$. Meanwhile, $\frac{po(f,C,\text{bounded})}{o(f,\text{bounded})} \leq 1$. Thus,

$$\begin{aligned} pr(< C, iMOBR >, f) &\leq \frac{po(f, C, \text{bounded}) + po(f, C, \text{bounding})}{o(f, \text{bounded}) + po(f, C, \text{bounding})} \\ &= \frac{po(f, C, \text{cMOBR})}{o(f, \text{bounded}) + po(f, C, \text{bounding})}. \end{aligned}$$

□

Based on the definition of the participation index, we can define the MaxPI bound as the smallest upper bound of the participation ratio of any feature in the local co-location pattern, i.e.,

$$\eta(< C, \text{cMOBR} >) = min_{f_i \in C}(\zeta(< C, \text{cMOBR} >, f_i)).$$

Given a participation index threshold θ, if $\eta(< C, \text{cMOBR} >) < \theta$, there will not be any eligible iMOBR in this cMOBR. In the pseudo-code, the MaxPI bound of C in every one of its cMOBRs, together with the number of instances, is compared with the thresholds to determine whether enumerating the iMOBRs in the current cMOBR is necessary.

Algorithm 2 Function MOBRGenerator in QGFR algorithm

Require:
CI: A set of instances of a co-location pattern C;
θ: Participation index threshold;
γ: Co-location instance number threshold;
l: The size of each grid cell.
Ensure: MOBRs of CI's subsets.

```
1: function MOBRGenerator(CI, θ, γ, l)
2:     AC ← active cells of C;
3:     for all subAC(with cardinality ≤ 4) ⊆ AC do
4:         cmobr ← the MOBR of subAC;
5:         η ← MaxPI(C, cmobr);
6:         n ← the number of C's instances in cmobr;
7:         if η ≥ θ and n ≥ γ then
8:             Add iMOBRs in cmobr to the result.
9:         end if
10:    end for
11: end function
```

Assuming that each co-location pattern has n_{ac} active cells on average, and the number of iMOBRs in each cMOBR is q, the computational complexity is $O(kn_{ac}^4 q)$. If q can be treated as a constant, because n_{ac} is much less than n_{ci} in most cases, the computational cost of the QGFR algorithm is much lower than that of the Quad. Because we have proved that in this algorithm all MOBRs of co-location instances are evaluated once and only eligible results are returned, we maintain the correctness and

completeness of the algorithm through the performance improvement.

2.4 Experimental Evaluation and Case Studies

In this section, we evaluate the baseline, Quad, and QGFR algorithm using synthetic data and a Chicago crime dataset [51], followed by one case study using the North American Atlas - Hydrography dataset from the U.S. Geological Survey [52] and the dataset of the U.S. major cities from Esri.

2.4.1 Experiments

The goal of the experiments was twofold: (a) evaluate the effect of the performance refinements of the proposed Quad algorithm and QGFR algorithm compared with the baseline algorithm. (b) determine the robustness of the QGFR algorithm given different inputs.

According to our analysis in §5.4, the computational complexity of the three algorithms are $O(k2^{n_{ci}}), O(kn_{ci}^4)$, and $O(kn_{ac}^4 q)$ respectively, where n_{ci} is the number of co-location instances per pattern, n_{ac} is the number of active cells per pattern, k is the number of co-location patterns, and q is the average number of iMOBR in each cMOBR. To evaluate the performance refinements, we studied the following two questions: (1) What is the effect of the number of co-location instances? (2) What is the effect of the number of co-location patterns? To determine the robustness, we asked how well the QGFR algorithm performed under different size of grid cells.

To answer these questions, we designed experiments as shown in Figure 2.5. The synthetic and the real-world data (a Chicago crime dataset) were generated with controlled parameters. In the simulation, three algorithms were executed with the grid cell size as a parameter. The performance was evaluated and compared using the run time of each algorithm. The platform for the simulation was Microsoft .NET Framework 4.5 on a computer with Intel(R) Core(TM) i7-4770 3.40 GHz CPU and 32 GB RAM. The parameters in the experiments are shown in Table 2.2.

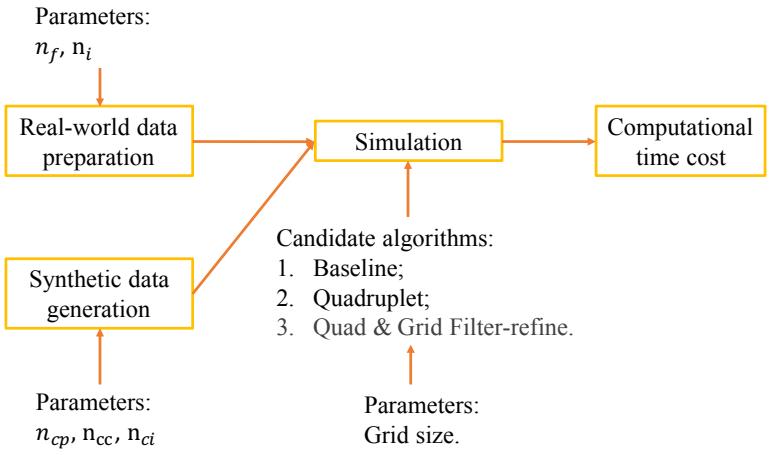


Figure 2.5: Experiment design.

Table 2.2: Parameters for the experiments.

Symbol	Meaning
n_{cp}	Number of core co-location patterns
n_{cc}	Core co-location patterns' cardinality
n_{ci}	Number of instances of each pattern
n_i	Number of input objects
n_f	Number of input features
Grid size	Cell's edge length of the grid used in the QGRF algorithm

Synthetic data generation

A point distribution with co-location patterns is often modeled as an aggregated point process [45, 53, 49]. Commonly used point processes include the Poisson cluster process [54] and Matérn's cluster process [55]. In order to ensure the existence of local co-location patterns, we made two changes on the steps used in [53], including:

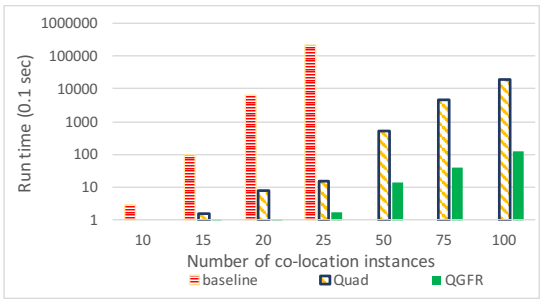
- Randomly select a rectangular region in the study area as a prevalence locality for each co-location pattern.
- In each co-location pattern's prevalence locality, ensure that at least 4 instances of the pattern are generated, and that no noise object of the features in the pattern is generated.

Because the subsets of a co-location pattern are also co-location patterns, when generating the synthetic data, we named the patterns which were not subsets of other patterns core patterns. The study area size was set to 10000×10000. The spatial relation was a neighborhood with a radius of 10. The number of noise objects of each feature was set to $4 \times n_{ci}$.

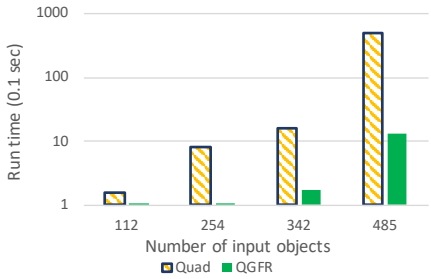
Experimental results

Effect of the number of co-location instances. The experiments were conducted with both synthetic and real-world data. The synthetic data was generated by fixing $n_{cp} = 2$ and $n_{cc} = 3$, but changing n_{ci}, whose results were shown in Figure 2.6a. The computational cost of the baseline algorithm, as expected, increased exponentially with n_{ci}, and was much larger than that of the two proposed algorithms, so its run time was not included when $n_{ci} = 50, 75,$ or 100. The run time of the Quad algorithm was much longer than that of the QGFR algorithm, and it also increased faster than the latter with increasing n_{ci}. The experiment with the Chicago crime dataset was conducted by fixing $n_f = 3$ but varying n_i. By increasing the number of input objects in a fixed study area, we increased the number co-location instances indirectly. The results (Figure 2.6b) also shown that the advantage of the QGFR algorithm increased as the number of input objects grew.

Effect of the number of co-location patterns. Since the number of co-location patterns is determined by both the number of core co-location patterns and their cardinalities, we conducted two controlled experiments with synthetic data and one with the Chicago



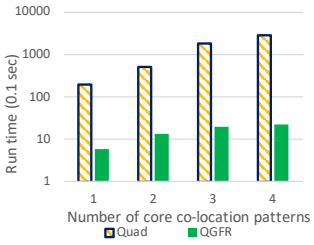
(a) Results with synthetic data.



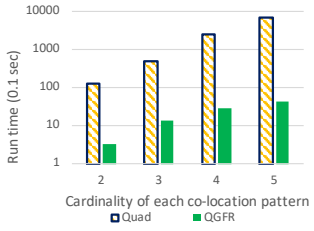
(b) Results with the Chicago crime dataset.

Figure 2.6: Effect of the number of co-location instances.

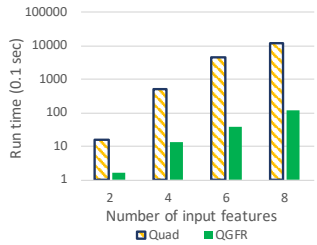
crime dataset on them. Figure 2.7a and Figure 2.7b presented the results of experiments with the synthetic data. In Figure 2.7a $n_{cc} = 3$ and $n_{ci} = 50$ but n_{cp} changed, while in Figure 2.7b $n_{cp} = 2$ and $n_{ci} = 50$ but n_{cc} changed. Figure 2.7c shown the results using the real-world data, where the number co-location pattern was increased by increasing the number of input features. In all the cases, the growing number of co-location patterns increased the advantage of the QGFR algorithm over the Quad algorithm.



(a) Effect of the number of core co-location patterns.



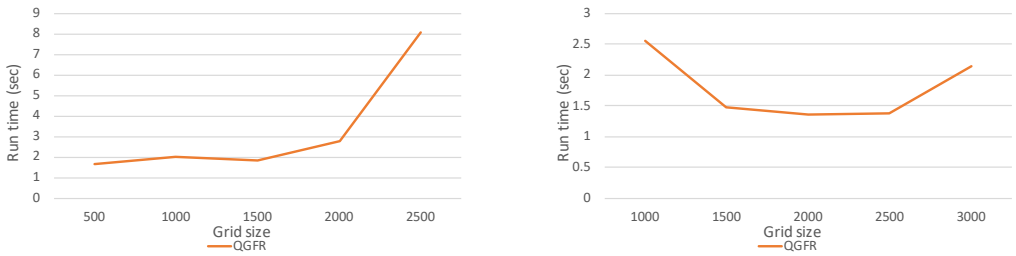
(b) Effect of each co-location pattern's cardinality.



(c) Effect of the number of input features.

Figure 2.7: Effect of the number of co-location patterns

Effect of the size of grid cells. The sensitivity analysis was done through two controlled experiments where the same synthetic and real-world data but different grid cell size were used. The parameters for the synthetic data were $n_{cp} = 2, n_{cc} = 3, n_{ci} = 50$ and those for real-world data were $n_i = 485, n_f = 4$. According to the results shown in Figure 2.8, the QGFR algorithm was robust with changes in the grid cell size, since the fluctuation of its run time was small when the grid cell size changed. When the grid cell size was small, the number of active cells was not much smaller than the number



(a) Results with synthetic data.

(b) Results with the Chicago crime dataset.

Figure 2.8: Effect of the size of grid cells.

of co-location instances, so the performance would be improved if a larger cell size was used. As the grid cell size increased, more iMOBRs resided in a single grid cell, so the performance improvement brought about by the MaxPI bound was weakened.

2.4.2 Case Study using North American Atlas-Hydrography and U.S. Major City Datasets

We conducted a case study using the North American Atlas - Hydrography dataset from the U.S. Geological Survey and the data of the U.S. major cities from Esri. Other inputs included a spatial relation specified by a neighborhood radius of 50 kilometers, a participation index threshold $\theta = 0.6$, and a instance number threshold $\gamma = 20$. There were 2610 cities which represent cities in the U.S. with population of more than 10 thousand in the dataset. The number of lakes was 394. The participation index of the co-location pattern $\{city, lake\}$ was 0.33, which meant major cities were not globally co-located with lakes in the U.S. However, our proposed QGFR algorithm detected some prevalence localities, two of which were shown in Figure 2.9 with the zoom-in maps. In the east locality, there were 163 cities, 109 of which were co-located with lakes, while 39 out of 41 lakes were near cities, so the participation index was about 0.67. This locality could be detected by the related work as well, because if we defined the density as the number of instances of a feature in a unit area, the density of both input objects and co-location instances was high (the ratio between the density of the co-location instances in the locality and that in the whole country was about 4.22). Contrarily, in the west locality, there were 35 out of 50 cities co-located with 7 out of 11 lakes, resulting in the participation index as about 0.63. In this locality, the density of the

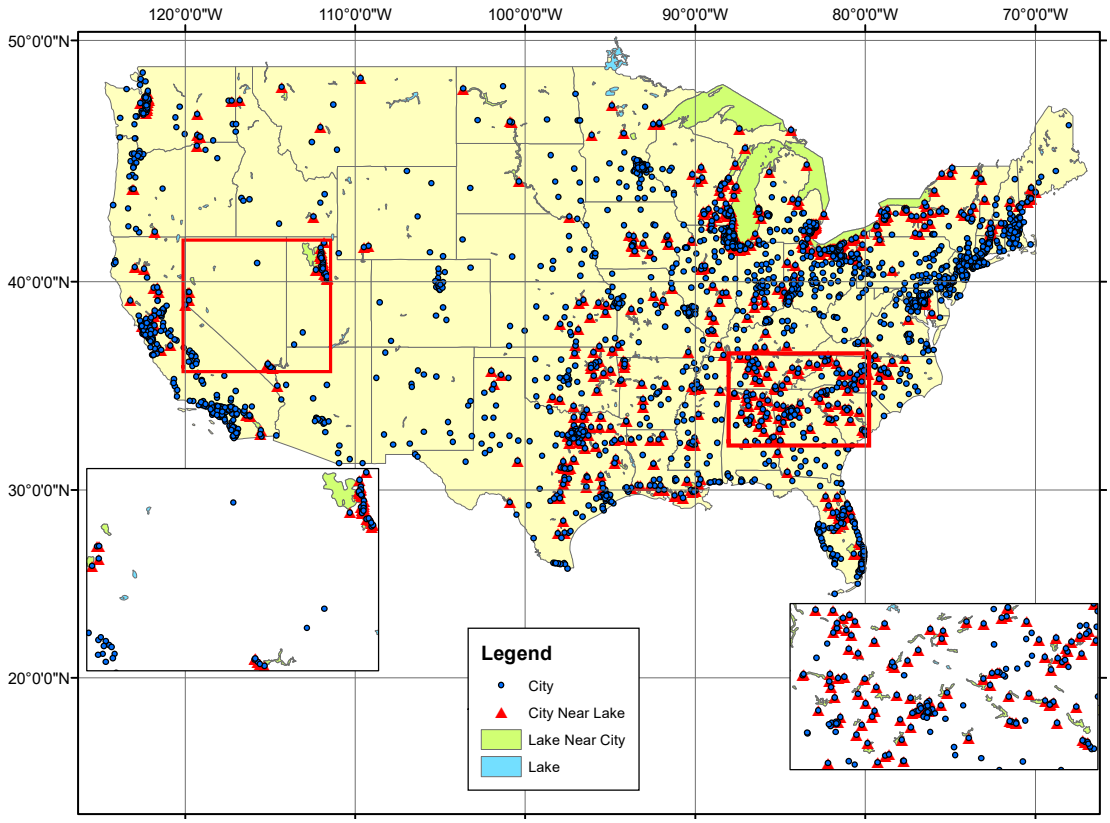


Figure 2.9: Case study with the hydrography and city data. Two prevalence localities of co-location pattern $\{city, lake\}$ are delineated by rectangles and shown in the zoom-in maps. (Better with color.)

input objects and co-location instances was almost the same as that in the whole country (the ratio between the density of the co-location instances in the locality and that in the whole country was about 1.03), which meant that the locality could not be identified by the related work using clustering to define localities. The findings indicated that the co-location pattern of major cities and lakes existed not only in the southeast of the U.S where lakes concentrated but also in the west where it was drier and lakes were more valuable of the cities.

2.5 Conclusion and Future Work

In this chapter, we formally defined the local co-location pattern detection problem, and proposed two algorithms that can efficiently solve it. The effectiveness and efficiency of the algorithms were proved theoretically and validated experimentally on synthetic and real datasets. In addition, we presented the results of one case study using the North American Atlas-Hydrography and U.S. Major City Datasets.

During the study, we noticed that the distribution of spatial events (e.g., the auto-correlation between events of the same feature) may affect the results. Our future research will take this into consideration. In addition, the distribution of events related to humans may be strongly affected by road networks especially in urban areas. Defining regions as subsets of road networks may result in richer and more meaningful results. We plan to explore this idea in our future work.

Chapter 3

Contrasting Spatial Colocation Pattern Detection

3.1 Introduction.

Contrasting spatial colocation pattern detection aims to discover spatial colocation patterns whose prevalence is substantially different in two groups of spatial datasets. Each spatial dataset contains objects belonging to different spatial features. For example, a spatial dataset may represent the points of interest in a city (e.g., hospitals, restaurants). A spatial colocation pattern refers to a set of spatial features, whose prevalence measures the tendency of any instances of the spatial features to be located near each other. For example, the set of McDonald's and Burger King restaurants is a prevalent colocation pattern in U.S. cities. However, this pattern may not have the same prevalence in a different spatial dataset. For example, Burger King and McDonald's are not commonly found near each other in China. Therefore, the goal of this problem is to find these kinds of colocation patterns given a prevalence difference threshold.

Applications domains: This problem is important for a variety of application domains. For example, in pathology, the distance between tumor cells and immune cells vary at different stages of diseases, which indicates their different relationships. Identifying contrasting spatial colocation patterns helps to generate hypotheses about the relationships between cells under different disease conditions, and also provide a novel way to diagnose diseases. Such patterns are common in other fields as well. In

environmental risk assessment, the colocation patterns of arsenic and other chemicals differ in water supply condition on the existence of pollution sites [56]. In wetland ecology, interspecies relationships vary significantly based on the growing environment, community composition, and species abundance [57]. Identifying contrasting spatial colocation patterns helps focus domain users' efforts on specific relationships that change with the underlying conditions.

Challenges: The challenges of the problem are two-fold. First, the number of potential patterns is exponentially related to the number of input spatial features. Second, the prevalence of spatial colocation patterns varies with the definitions of geographic proximity, so many definitions of geographic proximity have to be tested for each pattern.

3.1.1 Related Work and Limitations

Most studies on spatial colocation pattern detection originate from [58] where the concept of a spatial colocation pattern and its prevalence metric, the participation index, were introduced. Most research in this area falls into two groups. The first group aims to improve computational efficiency, such as the join-less [59], tree-based [60], and clique-based approaches [61, 62], as well as some parallel approaches [63, 64, 65]. The other group introduces variants of the participation index to achieve different detection objectives, such as patterns between extended objects [66] and fuzzy objects [67, 68], statistically significant patterns [69], and co-distribution patterns [70].

However, these studies all focus on detecting prevalent spatial colocation patterns in a single spatial dataset and cannot solve the problem in this paper efficiently. In general, the methods of spatial colocation pattern detection have two steps: (1) enumerating candidate spatial colocation patterns; and (2) generating the instances and calculating the prevalence of each pattern. All the studies on improving the computational efficiency of these methods focus on step (2), but leverage the anti-monotone property of the participation index to reduce the number of candidate patterns enumerated in step (1). According to this property, if a colocation pattern is not prevalent given a threshold, then its supersets are not prevalent either. In the contrasting spatial colocation pattern detection problem, since there is no prevalence threshold in the input, it is difficult for the anti-monotone property to act as a filter, and the number of candidate patterns in

step (1) becomes $2^{|F|}$, where $|F|$ is the number of input spatial features and can be large in many applications. For example, the number of cell types in a multiplexed pathology point set can be more than 30, and there are hundreds of different types of points of interest in a city, which makes the number of candidate patterns extremely large and the related work inapplicable.

3.1.2 Contributions

In this study, we introduce a metric to describe the difference between the prevalence of any spatial colocation patterns in two groups of spatial datasets, and a filter & refine algorithm to detect eligible contrasting spatial colocation patterns efficiently. Our contributions are summarized as follows.

- We formally define the problem of contrasting spatial colocation pattern detection.
- We introduce a metric to describe the difference between the prevalence of any spatial colocation patterns in two groups of spatial datasets. It is based on a commonly-used prevalence metric of spatial colocation patterns and satisfies the anti-monotone property that can be employed to enhance efficiency.
- We introduce a filter & refine algorithm utilizing the anti-monotone property of the proposed metric efficiently without affecting the completeness and correctness.
- We conduct extensive experiments which indicate that the proposed algorithm yields substantial computational time savings.
- We conduct a case study on a real-world dataset derived from multiplexed immunofluorescence images, which shows the capability of the proposed method to find patterns that are ignored by the related work, as well as the potential to advance scientific discovery.

Scope: We focus on contrasting spatial colocation patterns between two groups of spatial datasets. The generalization to more than two groups of spatial datasets is beyond the scope of this study. Generating colocation pattern instances and calculating the prevalence metric in a single spatial dataset, which is complementary with our study, is also beyond the scope.

3.2 Basic Concept and Problem Definition.

3.2.1 Basic concept.

A **spatial feature** refers to the conceptual abstraction of a set of spatial objects with the same feature type, such as a plant species or a business category. An **instance of spatial feature** f_i refers to a spatial object with feature type f_i. A **spatial object** is the representation of an entity or phenomenon in a 2/3-dimensional geographical space, e.g., a point representing a hospital or a tumor cell. A **spatial dataset** contains a set of spatial objects.

A **spatial colocation pattern** is defined as a set of spatial features. An **instance of spatial colocation pattern** C is a clique composed of one instance of each feature in C. A clique is a set of spatial objects among which every two objects are neighbors of each other given a neighbor relationship. For example, in spatial dataset (d) in Figure 3.1, $\langle f_a, f_b, f_c \rangle$ is a colocation pattern, and it has two instances represented by two sets of linked squares, circles and triangles.

A **neighbor relationship** exists between two spatial objects when the two objects are in geographic proximity. Geographic proximity in the spatial colocation pattern detection problem is typically defined using a distance threshold. Given a **neighbor distance threshold** d, two spatial objects are in geographic proximity if their relative distance is $\leq d$. Different neighbor distance threshold model neighbor relationships in different spatial resolutions. There are other ways of defining geographic proximity, but in this study, we only focus on the one using a distance threshold.

Given a neighbor distance threshold d, a commonly-used metric to measure the prevalence of a spatial colocation pattern C in a spatial dataset is the **participation index** ($\mathrm{PI}(C, d)$):

$$\mathrm{PI}(C, d) = \min_{\forall f_j \in C} \mathrm{PR}(C, f_j, d), \tag{3.1}$$

where $\mathrm{PR}(C, f_j, d)$ is the participation ratio of spatial feature f_j ($f_j \in C$) and $\mathrm{PR}(C, f_j, d) = \frac{|I(C, f_j, d)|}{|I(f_j)|}$, where $|I(C, f_j, d)|$ represents the number of unique instances of f_j in the instances of C, and $|I(f_j)|$ represents the number of all instances of f_j. The value of the participation index ranges from 0 to 1. The greater the value, the more prevalent the colocation pattern is. For example, in the spatial dataset in Figure 3.1, suppose that the neighbor relationship is defined by a threshold d. Then, $\mathrm{PR}(\langle f_b, f_c \rangle, f_b, d) = 0.75, \mathrm{PR}(\langle f_b, f_c \rangle, f_c, d) = 0.75$, and $\mathrm{PI}(\langle f_b, f_c \rangle, d) = 0.75$.

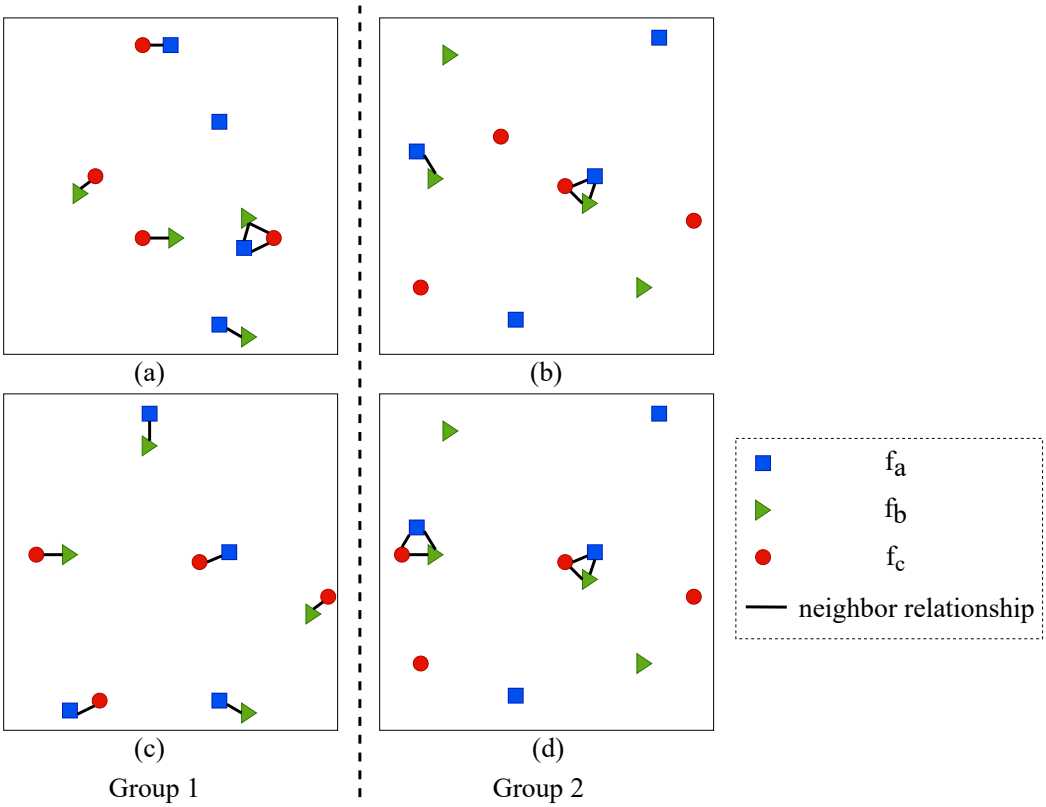


Figure 3.1: A sample input composed of four spatial datasets in two groups.

Given a neighbor distance threshold d, to measure the difference between the prevalence of a spatial colocation pattern C in two groups of spatial datasets, we propose a metric called the **participation index distribution difference** (PIDD(C, d)). The metric refers to the difference between the probability density functions of PI(C, d) in two groups of datasets, and is computed as follows. We use a histogram (denoted as $h_{C,d}(\cdot)$) to approximate the probability density functions of PI(C, d) in each group. For simplicity, in this study, the bins in all histograms have the same size, and the total area of a histogram is normalized to 1. For example, the PI$(\langle f_b, f_c\rangle, d)$ values in datasets (a) and (c) of Figure 3.1 are 0.75 and 0.5, respectively. Suppose that the number of bins in a histogram is 5. The probability density function of PI$(\langle f_b, f_c\rangle, d)$ of the datasets in group 1 can be represented by the left histogram in Figure 3.2. Similarly, the probability density function of PI$(\langle f_b, f_c\rangle, d)$ in group 2 can be represented by the right histogram. Then the difference between the two histograms is measured by the Manhattan distance between the vectors representing the bin areas of the histograms:

$$\text{PIDD}(C, d) = |h^{[1]}_{C,d} - h^{[2]}_{C,d}|_1. \tag{3.2}$$

For example, in Figure 3.2, $h^{[1]}_{\langle f_b,f_c\rangle,d} = [0, 0, 0.5, 0.5, 0]$, $h^{[2]}_{\langle f_b,f_c\rangle,d} = [0, 0.5, 0.5, , 0]$, so PIDD$(\langle f_b, f_c\rangle, d) = 1$. The value of participation index distribution difference ranges from 0 to 2. The larger the value, the larger the difference between the prevalence of a spatial colocation pattern of the two groups.

A **contrasting spatial colocation pattern** is defined as a spatial colocation pattern paired with a neighbor distance threshold, whose participation index distribution difference exceeds a given threshold.

To improve readability, a table of notations is provided in Table 3.1.

3.2.2 Problem definition.

The formal definition of the problem is as follows.

Input:

- A set of spatial features F;
- Two groups of spatial datasets containing the instances of F;
- A set of neighbor distance thresholds;

Table 3.1: Table of notations.

Notation	Meaning
f	A spatial feature.
C	A spatial colocation pattern.
d	A neighbor distance threshold.
$\mathrm{PI}(C, d)$	Given neighbor relationship defined by a neighbor distance threshold d, the participation index of a spatial colocation pattern C in a spatial dataset.
$\mathrm{PR}(C, f, d)$	Given neighbor relationship defined by a neighbor distance threshold d, the participation ratio of a spatial feature f in a spatial colocation pattern C in a spatial dataset.
$\mathrm{PIDD}(C, d)$	Given neighbor relationship defined by a neighbor distance threshold d, the participation index distribution difference of a spatial colocation pattern C in two groups of spatial datasets.
$h_{C,d}(\cdot)$	A histogram that represents the probability density function of $\mathrm{PI}(C, d)$ in a group of spatial datasets.
$H_{C,d}(\cdot)$	A histogram that represents the cumulative distribution function of $\mathrm{PI}(C, d)$ in a group of spatial datasets.

- The number of histogram bins;
- A participation index distribution difference threshold θ.

Output: Contrasting spatial colocation patterns.

3.3 Proposed Approach.

A baseline method to the problem has three steps: (1) traversing through the spatial colocation patterns; (2) for each colocation pattern, traversing through the neighbor distance thresholds; and (3) given each colocation pattern and distance threshold, generating instances of the pattern in all the input spatial datasets in two groups and computing the participation index distribution difference. All subsets of the input spatial features paired with certain distance thresholds may be contrasting spatial colocation patterns, which is exponentially related to the number of features. Hence, no matter how efficient the algorithm in step (3) is, a topic extensively studied in the related work, the time complexity of the solution is dominated by the number of patterns enumerated in step (1). Therefore, we propose the early-stop binary-search algorithm with two

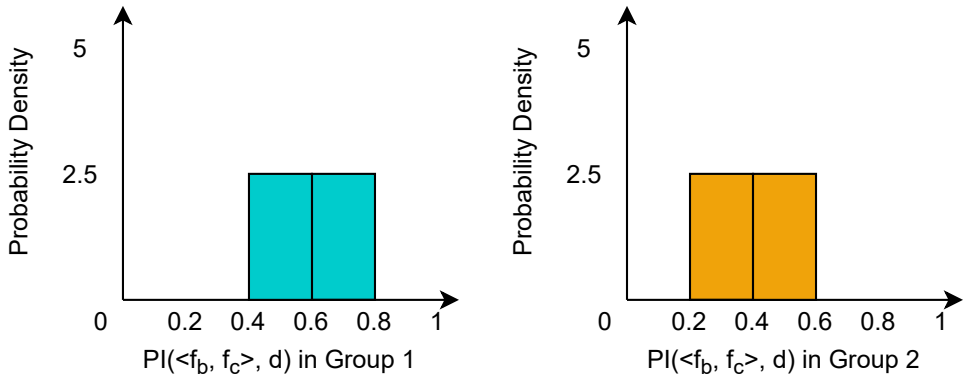


Figure 3.2: Histograms of $h_{\langle f_b,f_c\rangle,d}(\cdot)$ of two groups in Figure 3.1.

refinements: **binary distance threshold search** to reduce the number of neighbor distance thresholds enumerated in step (2), and **early-stop spatial colocation pattern enumeration** to reduce the number of colocation patterns enumerated in step (1).

3.3.1 Cumulative distribution function monotonicity.

The two refinements are based on the following two theorems.

Theorem 3.3.1. *Given a spatial colocation pattern C and two neighbor distance thresholds d_1 and d_2 where $d_1 \leq d_2$,*

$$H_{C,d_1}(b) \geq H_{C,d_2}(b), \forall b \in BIN, \tag{3.3}$$

where $H_{C,d_1}(\cdot)$ and $H_{C,d_2}(\cdot)$ are the histograms approximating the cumulative distribution functions of the $PI(C,d_1)$ and $PI(C,d_2)$ of a group of spatial datasets, respectively, and BIN is a set of bins in a histogram.

Theorem 3.3.2. *Given two spatial colocation patterns C_1 and C_2 where $C_1 \subset C_2$, and an neighbor distance threshold d,*

$$H_{C_1,d}(b) \leq H_{C_2,d}(b), \forall b \in BIN, \tag{3.4}$$

where $H_{C_1,d}(\cdot)$ and $H_{C_2,d}(\cdot)$ are the histograms approximating the cumulative distribution functions of the $PI(C_1,d)$ and $PI(C_2,d)$ of a group of spatial datasets, respectively, and

BIN is a set of bins in a histogram.

The proof of Theorem 3.3.1 and 3.3.2 is based on the following two Lemmas.

Lemma 3.3.3. *In a spatial dataset, $PI(C, d_1) \leq PI(C, d_2)$ if $d_1 \leq d_2$.*

Lemma 3.3.4. *In a spatial dataset, $PI(C_1, d) \geq PI(C_2, d)$ if $C_1 \subset C_2$.*

The proof of Lemma 3.3.3 is as follows.

Proof. According to the definition of an instance of a spatial colocation pattern, any instances of a pattern C with a neighbor distance threshold of d_1 must be the instances of C with a neighbor distance threshold of d_2, because $d_1 \leq d_2$. Therefore, for any spatial feature f in a colocation pattern C, the instances of f that are in the instances of C with a distance threshold of d_1 must be in the instances of C with a distance threshold of d_2, and $|I(C, f, d_1)| \leq |I(C, f, d_2)|$. According to the definition of the participation index,

$$\mathrm{PI}(C, d) = \min_{\forall f_j \in C} \mathrm{PR}(C, f_j, d),$$

where $\mathrm{PR}(C, f_j, d) = \frac{|I(C, f_j, d)|}{|I(f_j)|}$, Lemma 3.1 is proved. □

Based on Lemma 3.3.3, Theorem 3.3.1 is proved as follows.

Proof. Because of Lemma 3.1, in a group of spatial datasets, the number of datasets with $\mathrm{PI}(C, d_1) \leq x$ is no less than the numbers with $\mathrm{PI}(C, d_2) \leq x, \forall x$. $H_{C,d}(b)$ is the histogram approximating the cumulative distribution function of $\mathrm{PI}(C, d)$ evaluated at b, which refers to the probability that $\mathrm{PI}(C, d)$ will take a value less than or equal to b, so $H_{C,d_1}(b) \geq H_{C,d_2}(b), \forall b \in BIN$. □

The proof of Lemma 3.3.4 is as follows.

Proof. According to the definition of an instance of a spatial colocation pattern, any instances of a pattern C_2 must contain an instance of a pattern C_1 with the same neighbor distance threshold d, because $C_1 \subset C_2$. Therefore, for any spatial feature f in a colocation pattern C, the instances of f that are in the instances of C_2 must also be in the instances of C_1, and $|I(C_1, f, d)| \geq |I(C_2, f, d)|$. According to the definition of the participation index, Lemma 3.2 is proved. □

Then, Theorem 3.3.2 is proved as follows.

Proof. Because of Lemma 3.2, in a group of spatial datasets, the number of datasets with $\mathrm{PI}(C_1, d) \leq x$ is less than or equal to the numbers with $\mathrm{PI}(C_2, d) \leq x, \forall x$. $H_{C,d}(b)$ is the histogram approximating the cumulative distribution function of $\mathrm{PI}(C, d)$ evaluated at b, which refers to the probability that $\mathrm{PI}(C, d)$ will take a value less than or equal to b, so $H_{C,d_1}(b) \leq H_{C,d_2}(b), \forall b \in BIN$. □

3.3.2 Binary distance threshold search.

For a spatial colocation pattern C, when the baseline method searches for any neighbor distance thresholds d in a range $[d_1, d_2]$ such that $\mathrm{PIDD}(C, d)$ exceeds a given threshold, it traverses through all the distance thresholds in $[d_1, d_2]$. The key idea of this refinement is that if we can prove that the lower bound of $\mathrm{PIDD}(C, d)$ exceeds the threshold or that the upper bound of $\mathrm{PIDD}(C, d)$ is less than the threshold, for all $d \in [d_1, d_2]$, we can reduce the number of distance thresholds enumerated.

Let's denote the histograms approximating the cumulative distribution function of $\mathrm{PI}(C, d)$ in two groups of spatial datasets as $H^{[1]}_{C,d}$ and $H^{[2]}_{C,d}$, respectively. Then, according to Equation (3.2), the participation index distribution difference $\mathrm{PIDD}(C, d)$ is

$$\Sigma_{i=1}^{k} |(H^{[1]}_{C,d}(b_i) - H^{[1]}_{C,d}(b_{i-1})) - (H^{[2]}_{C,d}(b_i) - H^{[2]}_{C,d}(b_{i-1}))|, \tag{3.5}$$

where b_i is the ith bin in a histogram with k bins, and $H_{C,d}(b_0) = 0$. The task of finding the upper and lower bounds of $\mathrm{PIDD}(C, d), \forall d \in [d_1, d_2]$ given the H_{C,d_1}, H_{C,d_2} of two groups of spatial datasets is formalized as the following optimization problem.

$$\begin{aligned}
&\max_{H^{[1]}_{C,d}, H^{[2]}_{C,d}} \mathrm{PIDD}(C, d) \\
\text{and } &\min_{H^{[1]}_{C,d}, H^{[2]}_{C,d}} \mathrm{PIDD}(C, d) \\
\text{subject to: } &H^{[j]}_{C,d_1}(b_i) \geq H^{[j]}_{C,d}(b_i) \geq H^{[j]}_{C,d_2}(b_i), && (3.6)\\
&H^{[j]}_{C,d}(b_{i-1}) \leq H^{[j]}_{C,d}(b_i), && (3.7)\\
&j = 1, 2, \text{ and } \forall b_i \in BIN.
\end{aligned}$$

Constraint (3.6) is based on Theorem 3.3.1. Constraint (3.7) is true because cumulative distribution functions are monotonic increasing. According to Equation (3.5), the objective function is a linear function, so the optimum is attained on the vertices of the feasible region. Searching for the optimums by enumerating the vertices is equivalent to generating a path of $\mathbb{H} = [(H^{[1]}_{C,d}(b_1),H^{[2]}_{C,d}(b_1)), ..., (H^{[1]}_{C,d}(b_k),H^{[2]}_{C,d}(b_k))]$, where the values of $H^{[j]}_{C,d}(b_i)$ are $H^{[j]}_{C,d_1}(b_l)$ and $H^{[j]}_{C,d_2}(b_l)$ that are in the range $[H^{[j]}_{C,d_2}(b_i), H^{[j]}_{C,d_1}(b_i)]$ for $l = 1, 2, ..., k$, which we call its vertex values. In the worst case, the numbers of vertex values of $(H^{[1]}_{C,d}(b_i)$ and $H^{[2]}_{d}(C, b_i))$ are both $2k$, so the number of possible values of the path $\mathbb{H}$ is $(2k)^{2k}$, making the process of searching for the optimum computationally expensive.

To reduce this cost, we introduce a dynamic-programming algorithm, called CumDP, based on the Markov property of the feasible region of $H^{[j]}_{C,d}(b_i)$. According to this property condition on $H^{[j]}_{C,d}(b_{i-1})$, the feasible region of $H^{[j]}_{C,d}(b_i)$ is irrelevant to $H^{[j]}_{C,d}(b_m)$ where $m < i - 1$. Denote

$$\begin{aligned} \text{PIDD}(C, d)[: t] =& \Sigma^{t}_{i=1} |(H^{[1]}_{C,d}(b_i) - H^{[1]}_{C,d}(b_{i-1})) \\ & - (H^{[2]}_{C,d}(b_i) - H^{[2]}_{C,d}(b_{i-1}))|, \end{aligned} \tag{3.8}$$

where $t = 1, 2, ..., k$. This gives us the following theorem.

Theorem 3.3.5. *Denote the optimum (maximum or minimum) of the PIDD*$(C, d)[: t]$ *condition on* $H^{[1]}_{C,d}(b_t) = \alpha_t$, $H^{[2]}_{C,d}(b_t) = \beta_t$ *as* $\bar{\Delta}_t(\alpha_t, \beta_t)$. *Then,*

$$\begin{aligned} \bar{\Delta}_t(\alpha_t, \beta_t) = \underset{\alpha_{t-1}, \beta_{t-1}}{\textit{Optimum}} & (\bar{\Delta}_{t-1}(\alpha_{t-1}, \beta_{t-1}) + \\ & |(\alpha_t - \beta_t) - (\alpha_{t-1} - \beta_{t-1})|), \end{aligned} \tag{3.9}$$

where α_t, β_t *are the vertex values of* $H^{[1]}_{C,d}(b_t)$ *and* $H^{[2]}_{C,d}(b_t)$, *respectively, and* $\alpha_t \geq \alpha_{t-1}$, *and* $\beta_t \geq \beta_{t-1}$.

The proof of this theorem is as follows.

Proof. Since $\text{PIDD}(C, d)[: t]$ is a linear function of $H^{[1]}_{C,d}(b_t)$, $H^{[2]}_{C,d}(b_t)$, $H^{[1]}_{C,d}(b_{t-1})$ and $H^{[2]}_{C,d}(b_{t-1})$, its optimums are reached when they are at their vertex values. In addition, condition on $H^{[j]}_{C,d}(b_{t-1})$, the feasible region of $H^{[j]}_{C,d}(b_i)$ is irrelevant to $H^{[j]}_{C,d}(b_m)$ where

$m < i - 1$. Therefore, the optimums of PIDD$(C, d)[: t]$ can be found by enumerating the combinations of the vertex values of $H^{[1]}_{C,d}(b_t)$, $H^{[2]}_{C,d}(b_t)$, $H^{[1]}_{C,d}(b_{t-1})$ and $H^{[2]}_{C,d}(b_{t-1})$ as well as the corresponding optimums of PIDD$(C, d)[: t - 1]$. □

Based on Theorem 3.3.5, the CumDP algorithm consists of the following two steps.

1. **Initialization:** Set $\bar{\Delta}_0(\cdot, \cdot) = 0$ for all vertex values of $H^{[1]}_{C,d}(b_0)$ and $H^{[2]}_{C,d}(b_0)$.
2. **Updating:** Compute the values of $\bar{\Delta}_t(\alpha_t, \beta_t)$ using Equation (3.9) until $t = k$.

The CumDP algorithm enumerates pairs of possible values of (α_t, β_t) and $(\alpha_{t-1}, \beta_{t-1})$ for $t = 1, 2, ..., k$. Thus, the time complexity of the algorithm is $O(k^5)$, which is much smaller than the baseline algorithm ($O(k^k)$).

In sum, we have an efficient algorithm to determine the optimums of PIDD(C, d) given the H_{C,d_1}, H_{C,d_2} of two groups of spatial datasets for $d \in [d_1, d_2]$. Now we can replace the linear enumeration of the neighbor distance thresholds in step (2) of the baseline method with the **binary distance threshold search** algorithm. The idea of this algorithm is that for a spatial colocation pattern C, the process of searching for any neighbor distance thresholds d in a range $[d_{lo}, d_{hi}]$ so that PIDD$(C, d) \geq \theta$ is composed of the following three steps.

1. Compute PIDD(C, d_{lo}) and PIDD(C, d_{hi}), and record the $H_{C,d_{lo}}$, $H_{C,d_{hi}}$ of the two groups of spatial datasets.
2. Use the $H_{C,d_{lo}}$, $H_{C,d_{hi}}$ of two groups of spatial datasets to determine the maximum and minimum of PIDD(C, d) where $d \in [d_{lo}, d_{hi}]$.
3. If the maximum of PIDD$(C, d) < \theta$, no neighbor distance thresholds in the range $[d_{lo}, d_{hi}]$ is eligible. If the minimum of PIDD$(C, d) \geq \theta$, all distance thresholds in the range are eligible. Otherwise, divide the range $[d_{lo}, d_{hi}]$ equally into two parts: $[d_{lo}, d_{mid}]$ and $[d_{mid}, d_{hi}]$. And recursively search for eligible distance thresholds in the two ranges.

The detailed pseudocode is in Algorithm 3.

3.3.3 Early-stop spatial colocation pattern enumeration.

Our second algorithmic refinement, early-stop spatial colocation pattern enumeration, is based on the idea that by keeping track of the set of spatial colocation patterns whose supersets may be eligible, we can avoid enumerating the patterns whose subsets are not

Algorithm 3 Binary distance threshold search (BDTS) algorithm

Require:
C: a spatial colocation pattern;
D: a sorted list of neighbor distance thresholds;
lo, hi: the index of the first and the last distance thresholds to be considered;
Mem: the storage to save the histograms of the cumulative distribution functions of the participation index of C in two groups of spatial datasets;
θ: A participation index distribution difference thresholds.
Ensure: Contrasting spatial colocation patterns.

```
ans ← [].
Get H_{d_lo}^{[j]} and H_{d_hi}^{[j]} from Mem where j = 1, 2. If an entry does not exist in Mem, calculate and save it in Mem.
max, min ← the optimums of PIDD(C, d) for d ∈ [d_lo, d_hi] using the CumDP algorithm.
if min ≥ θ then
    for d ∈ [d_lo, d_hi] do
        ans.add((C, d)).
    end for
else if max ≥ θ then
    mid ← (lo + hi)/2;
    firstAns ← Recursively call this algorithm BDTS(C, D, lo, mid, Mem, θ).
    secondAns ← Recursively call this algorithm BDTS(C, D, mid, hi, Mem, θ).
    ans ← firstAns + secondAns
end if
return ans
```

in the set. Once the set is empty, we can stop enumerating spatial colocation patterns.

Suppose we know $H^{[1]}_{C_1,d}$ and $H^{[2]}_{C_1,d}$, which are histograms that approximate the cumulative distribution functions of the $\mathrm{PI}(C_1, d)$ in two groups of spatial datasets. Given a neighbor distance threshold d, if we can prove that an upper bound of $\mathrm{PIDD}(C, d)$ is less than the input participation index distribution difference threshold $\forall C \supset C_1$, we can avoid enumerating the supersets of C_1, since none of them would be eligible.

The task of finding the upper bound of PIDD(C, d) is formalized as follows:

$$\begin{aligned} \underset{H^{[1]}_{C,d}, H^{[2]}_{C,d}}{\text{Max}} \quad & \text{PIDD}(C, d) \\ \text{subject to:} \quad & H^{[j]}_{C,d}(b_i) \geq H^{[j]}_{C_1,d}(b_i), && (3.10) \\ & H^{[j]}_{C,d}(b_{i-1}) \leq H^{[j]}_{C,d}(b_i), && (3.11) \\ & j = 1, 2, \text{ and } \forall b_i \in BIN. \end{aligned}$$

Constraint (3.10) is based on Theorem 3.3.2, and Constraint (3.11) is true because cumulative distribution functions are monotonic increasing. This optimization problem is similar to the one we solved in Section 3.3.2. The only difference is that the feasible region of $H^{[j]}_{C,d}(b_i)$ specified by Constraint (3.10) has no upper bound, while that of $H^{[j]}_{C,d}(b_i)$ specified by Constraint (3.6) does. If we replace Constraint (3.10) with

$$1 \geq H^{[j]}_{C,d}(b_i) \geq H^{[j]}_{C_1,d}(b_i), \tag{3.12}$$

because the value of a cumulative distribution function at any point should be within the range $[0, 1]$, we can apply the CumDP algorithm and get the same optimum we get in the original problem.

However, in order to determine whether we can avoid enumerating the supersets of a colocation pattern C in the current settings, we have to know $H^{[1]}_{C,d}$ and $H^{[2]}_{C,d}$ for all neighbor distance thresholds. It cannot be applied with the binary distance threshold search refinement proposed in Section 3.3.2 whose purpose is to avoid computing the $H^{[1]}_{C,d}$ and $H^{[2]}_{C,d}$ of all neighbor distance thresholds. In order to resolve this conflict, we loosen Constraint (3.12) by replacing it with

$$1 \geq H^{[j]}_{C,d}(b_i) \geq H^{[j]}_{C_1,d_\infty}(b_i), \tag{3.13}$$

where d_∞ is the longest input neighbor distance threshold. Because $d \leq d_\infty$, $H^{[j]}_{C_1,d}(b_i) \geq H^{[j]}_{C_1,d_\infty}(b_i)$ according to Theorem 3.3.1. Therefore, the feasible region specified by Constraint (3.12) is smaller than that specified by Constraint (3.13), and the maximum we get with Constraint (3.13) must be no less than the maximum we get with Constraint (3.12) and can be used as an upper bound. In this way, for each colocation pattern C, we

only need to know $H^{[1]}_{C,d_\infty}$ and $H^{[2]}_{C,d_\infty}$ to determine whether its supersets may be eligible.

In this refinement, spatial colocation patterns are enumerated in ascending order by their cardinality. To determine whether a spatial colocation pattern with n spatial features needs to be enumerated, we only need to check whether all of its subsets with $n-1$ spatial features are in the set of patterns whose supersets may be eligible. If not, this pattern can be ignored.

3.4 Experiments.

The goal of the experiments was twofold: (a) a *comparative analysis* to evaluate the effect of the two refinements of the early-stop binary-search (ESBS) algorithm on computational time cost and (b) a *sensitivity analysis* to determine the scalability of the ESBS algorithm.

For the comparative analysis, we asked whether the proposed refinements yield computational time savings. The following candidate algorithms were included in the analysis:

- The baseline algorithm (Baseline)
- The algorithm with the binary distance threshold search refinement (Binary)
- The algorithm with the early-stop spatial colocation pattern enumeration refinement (Early-stop)
- The ESBS algorithm (ESBS)

Since the methods of generating spatial colocation pattern instances in a single spatial dataset in the related work are complementary with our study, for simplicity, we used the join-less method proposed in [59] in the experiments. This method can be easily replaced by other methods and does not affect the experiment results. The sensitivity analysis evaluated the effect of the number of spatial features $|F|$, the number of neighbor distance thresholds $|D|$, the input participation index distribution difference threshold θ, and the number of spatial datasets $|P|$ on the computational time of ESBS. Table 3.2 shows the detailed parameters in the experiments.

The experimental design is illustrated in Figure 3.3. Each set of experiments was repeated ten times. Observed execution time was the metric of computational time cost. Experiments were performed on a computer with a quad-core Intel(R) Xeon(R) CPU E5-2623 v3 (3.00GHz), 64GB memory, and Python 3.7.

Table 3.2: Experiment parameters for the sensitivity analysis.

Experiment (Effect of)	$\|D\|$	$\|F\|$	θ	$\|P\|$
$\|D\|$	Vary	3	0.8	200
$\|F\|$	5	Vary	1	200
θ	10	4	Vary	200
$\|P\|$	10	4	1	Vary

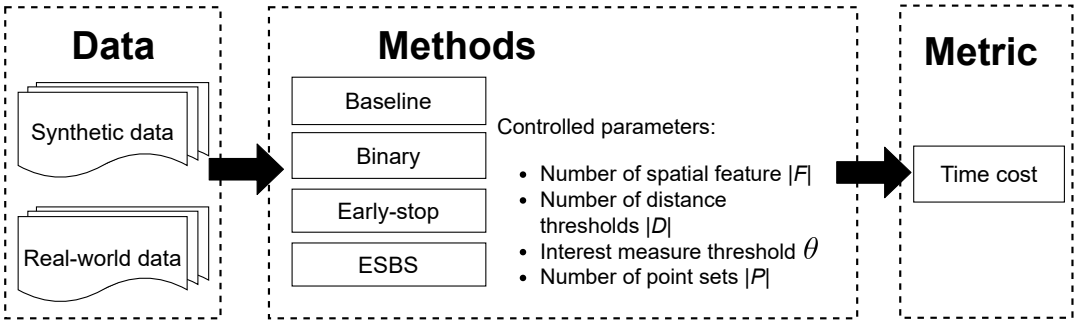


Figure 3.3: Experiment design.

3.4.1 Datasets.

Experiments were conducted on both synthetic and real-world data.

The synthetic data was generated under the assumption that the distribution of the spatial objects in the spatial datasets in two groups obey the same homogeneous Poisson process (i.e., complete spatial randomness). The number of spatial objects in one spatial feature in a spatial dataset was set to 100. The size of the study area was 1000×1000.

The real-world data was generated from a collection of pathology spatial datasets belonging to two disease groups. A pathology spatial dataset contained points that represented the locations and types of cells in an multiplexed immuno-fluorescence image. In the original data, there were nine types of cells, and 56 and 143 datasets in two groups, respectively. When generating data with $|F|$ spatial features and $|P|$ datasets, we sampled with replacement $|P|$ datasets and sampled without replacement points representing $|F|$ types of cells in each dataset.

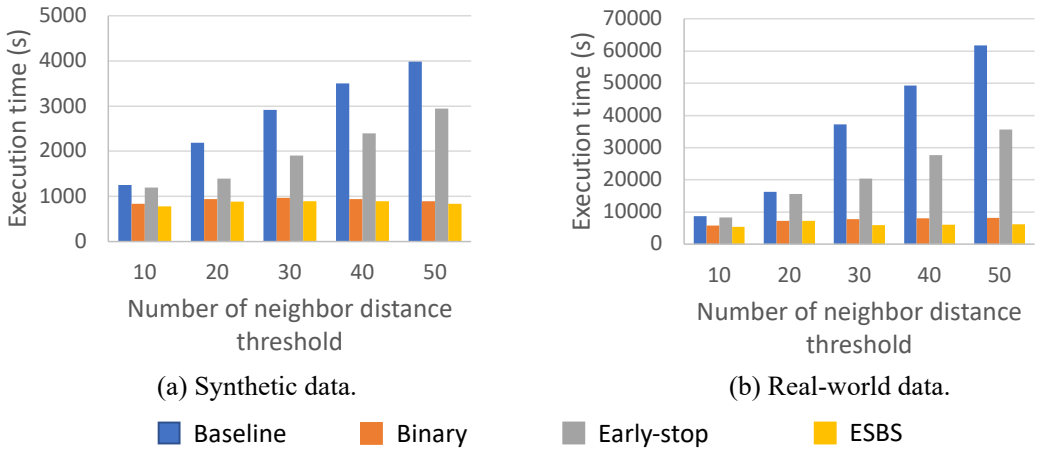


Figure 3.4: Effect of the number of neighbor distance thresholds.

3.4.2 Experiment Results.

Effect of the number of neighbor distance thresholds. In this set of experiments, the number of neighbor distance thresholds varied from 10 to 50. The results of Baseline (blue) and Binary (orange) in Figure 3.4 indicate the binary spatial distance threshold search refinement reduced the computational cost significantly. The results of Early-stop (gray) and ESBS (yellow) shows a similar trend, even though the early-stop refinement was in effect and moderated the advantage of ESBS over early-stop. The computational time savings generated by the binary spatial distance threshold search refinement increased with the number of neighbor distance thresholds.

Effect of the number of input spatial features. In this set of experiments, the number of input spatial feature types was set to 3, 4, and 5. Figure 3.5 shows that the early-stop spatial colocation pattern enumeration refinement reduced the computational time cost significantly. As the number of input spatial features increased, the advantages of the algorithms with the refinement became more significant. In addition, the results in Figure 3.4 and 3.5 indicate that when the number of input spatial features was small (e.g., 3) the computational time savings was due to the binary distance threshold search refinement, while when the number of input spatial features increased, it was the early-stop refinement that became important.

Effect of PIDD threshold. In this set of experiments, the PIDD threshold was set as

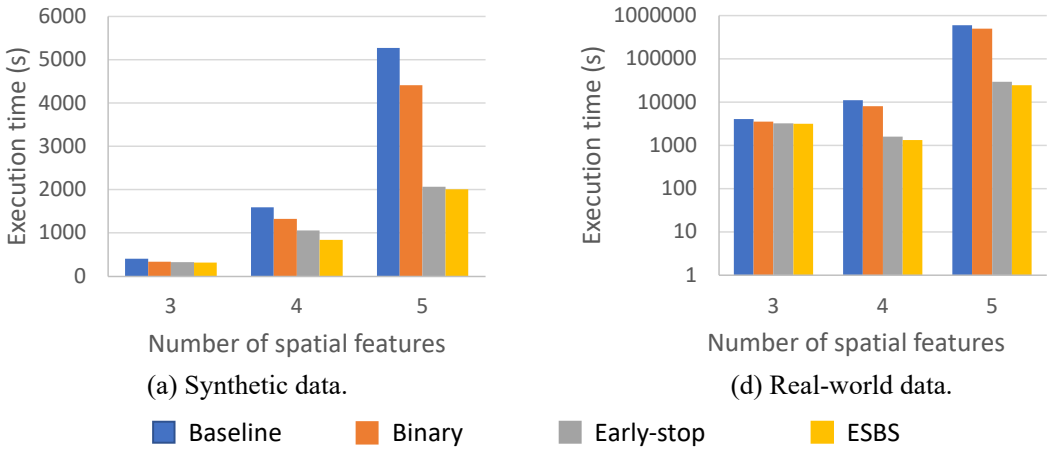


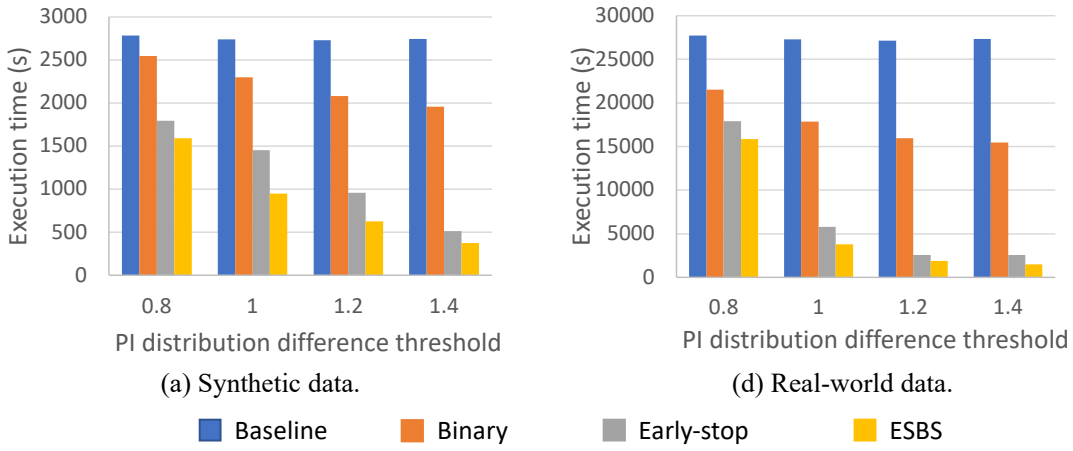
Figure 3.5: Effect of the number of spatial features.

0.8, 1, 1.2, 1.4. The results show that the effect of both refinements depended heavily on the value of the threshold. When the threshold was small (e.g, 0.8), the ESBS algorithm only yielded 43% execution time savings compared with the baseline algorithm in real-world data experiments. When the threshold was large (e.g., 1.4), the ESBS algorithm yielded more than 95% execution time savings.

Effect of number of spatial datasets. We varied the number of spatial datasets from 100 to 250. The results (Figure 3.7) indicated that the execution time was almost linearly related to the number of datasets. This was reasonable because the proposed algorithmic refinements did not affect the generation of spatial colocation pattern instances in each dataset, and when calculating the participation index distribution difference of a spatial colocation pattern given a neighbor distance threshold, every spatial dataset has to be visited once.

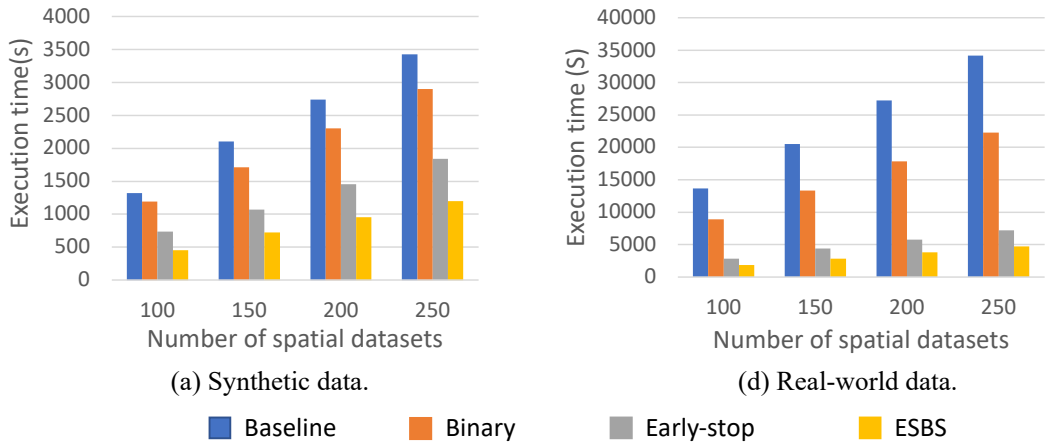
3.5 Case Study

The case study was conducted on the real-world data that we used in the experiments in Section 3.4, which contained spatial datasets from multiplexed immuno-fluorescence images. A spatial dataset contains points that record the locations and attributes (e.g., surface phenotype markers) of the cells in the image of a tissue sample, which becomes



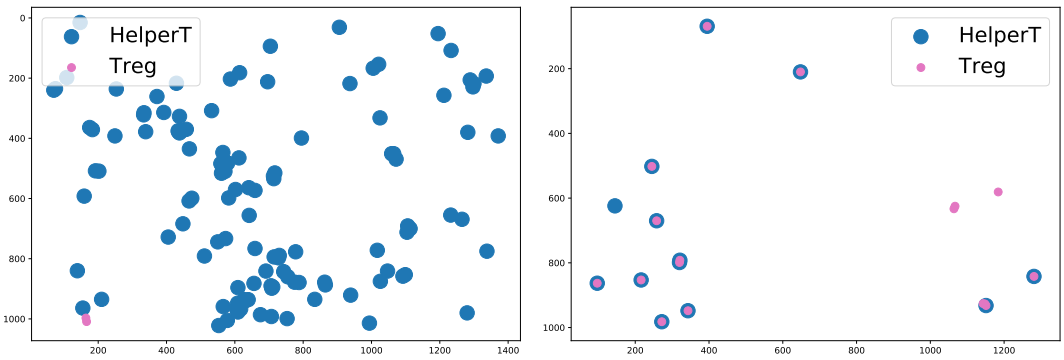
(a) Synthetic data. (d) Real-world data.

Figure 3.6: Effect of PI distribution difference threshold.



(a) Synthetic data. (d) Real-world data.

Figure 3.7: Effect of number of spatial datasets.



(a) Chronic pancretitis (b) Pancreatic ductal adenocarcinoma

Figure 3.8: Distribution of HelperT and Treg cells in the sample spatial datasets of two groups.

possible recently because of the developments in whole slide digital imaging and antigen-based staining technology. Representing types and locations of cells with points facilitates understanding the interplay between various cells in a spatially informed manner in the tumor micro-environment [31, 32]. Currently, visual inspection and cell-counting by a pathologist are the methods used to quantify the different phenotype of cells present in the tissue micro-environment. However, this practice is fraught with inter-observer variability and inconsistency between studies.

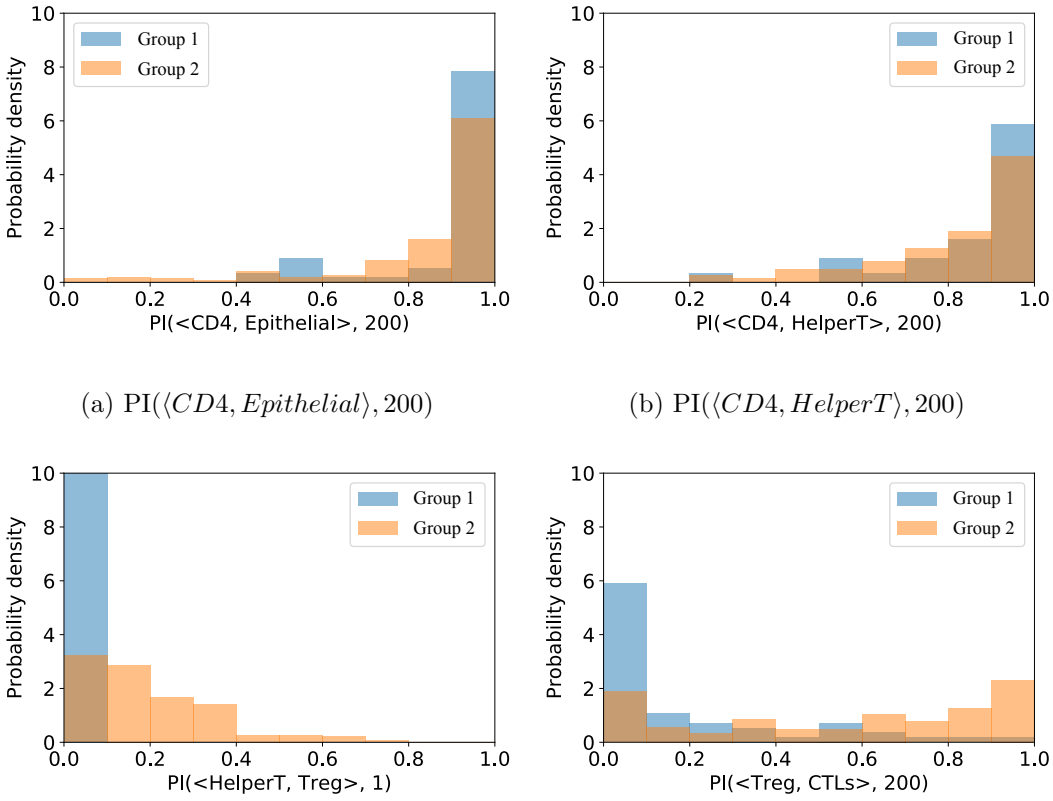
The proposed contrasting spatial colocation pattern detection method provides a novel way to characterize the interplay between cells under different disease conditions. The spatial datasets used in this case study belonged to two disease groups, namely, chronic pancretitis (Group 1) and pancreatic ductal adenocarcinoma (Group 2). There were nine types of cell surface phenotype markers: Treg, APC, Epithelial, HelperT, PDL1_CD3, PDL1_CD8, PDL1_FoxP3, CD4, and CTLs. The participation index distribution difference threshold was set to 1.2. The set of neighbor distance thresholds was $[1, 50, 100, 150, 200]$. The number of histogram bins was 10.

An example of contasting spatial coloation patterns detected by the proposed method is $(\langle HelperT, Treg \rangle, 1)$. Figure 3.8 shows the spatial distribution of HelperT and Treg cells in two spatial datasets. As can be seen, in the Group 2 dataset HelperT cells tend to overlap with Treg cells, while in the Group 1 dataset they do not. This observation is also supported by the histograms in Figure 3.9c, where most of the

$\text{PI}(\langle HelperT, Treg\rangle, 1)$ values in Group 1 were 0, while in Group 2 datasets, there were cases when $\text{PI}(\langle HelperT, Treg\rangle, 1) > 0$. Hence, the proposed method successfully distinguished the spatial neighborhood relationships between HelperT and Treg cells in the two groups. An explanation for this phenomenon is as follows. Treg cells have a regulating effect on the immune response of the locale [71]. In the cancer micro-environment (Pancreatic ductal adenocarcinoma), a large portion of the HelperT cells are inhibited by the Treg cells colocated with them, which may be a result of or a cause to the cancer.

If we apply the spatial colocation pattern detection methods in the related work, which focus on finding prevalent patterns in a single spatial dataset, we have to first define the prevalence of a spatial colocation pattern in a group of spatial datasets. Suppose given a neighbor distance threshold, we define a prevalent spatial colocation pattern as the one whose probability of getting a participation index exceeding 0.6 is greater than or equal to 70%. A related work would yield pairs such as $(\langle CD4, Epithelial\rangle, 200)$ and $(\langle CD4, HelperT\rangle, 200)$. By comparing the histograms of the probability density functions of the PIs of these patterns (Figure 3.9 (a) (b)) with those of the patterns found by the proposed method (Figure 3.9 (c) (d)), we can find that the proposed method can find pairs of spatial colocation patterns and neighbor distance thresholds whose PIs tend to be different while the existing methods cannot.

From a clinical perspective, the results highlight some key cell relationships that may directly or indirectly play a role in the disease micro-environment. Specifically, the relationship between CTLs and Treg, and HelperT and Treg are of particular interest from an immunological standpoint. CTLs (Cytotoxic Lymphocytes) are cells that actively seek out and kill cancer cells in the environment on activation of the immune system [72]. Under normal conditions, the Treg cells have a regulating effect on the immune response of the locale [71]. In the cancer micro-environment, however, it has been observed that Treg cells play a more functional role. As shown in Figure 3.9 (d), in the multiplexed immuno-fluorescence images of the pancreatic ductal adenocarcinoma group (Group 2), Treg cells tend to be colocated with CTLs cells, which potentially inhibits the function of CTLs cells [73]. This may be due to physiologic suppression of activated CTLs, or pathological polarization of CD4 positive cells by tumor secreted factors in the tumor micro-environment[74]. Further investigation on a larger cohort to confirm the potential



(a) PI($\langle CD4, Epithelial \rangle, 200$)

(b) PI($\langle CD4, HelperT \rangle, 200$)

(c) PI($\langle HelperT, Treg \rangle$, 1)

(d) PI($\langle Treg, CTLs \rangle, 200$)

Figure 3.9: The probability density functions of PIs of sample colocation patterns and distance thresholds pairs detected by a related work ((a), (b)) and those detected by the proposed method ((c), (d)).

discriminatory power of the pairwise interactions observed in this experiment would be warranted.

3.6 Conclusion & Future Work

This work explored the problem of contrasting spatial colocation pattern detection in relation to application domains such as pathology, environment risk assessment, and wetland ecology. An early-stop binary-search algorithm is proposed which discovers contrasting spatial colocation patterns whose participation index distribution difference exceed a given threshold in two groups of spatial datasets. The proposed approach uses upper and lower bound pruning as well as algorithmic refinements to enhance its scalability. Experimental evaluation indicated that the proposed refinements yield substantial computational time savings. A case study on a real-world pathology dataset presented the method's potential to support scientific discovery.

In the future, we plan to generalize the problem to more than two groups of spatial datasets, as well as spatial relationships other than spatial colocation.

Chapter 4

SRNet: A spatial-relationship aware point-set classification method for multiplexed pathology images

4.1 Introduction

Point-set classification for multiplexed pathology images aims to distinguish between the spatial configurations of cells within multiplexed immuno-fluorescence (mIF) images of different diseases. Advances in the field of multiplexed and anti-body based imaging methods have promoted the development of mIF images, which facilitates bio marker-specific cell species and sub species identification [75]. An example of a multipled immunoflourescene image is show in Figure 4.1. A point set from multiplexed pathology images records the location and the attributes (e.g., surface phenotype markers) of the cells in a mIF image. For example, Figure 4.2 shows a sample point set from a mIF image. The location of each cell is represented by its pixel coordinates whose origin is at the top left corner of the image. The cell attributes are the existence of surface phenotype markers (e.g., Epithelial), where "pos" means the presence of a phenotype marker and "neg" otherwise. Figure 4.3 illustrates the spatial distribution of "pos"

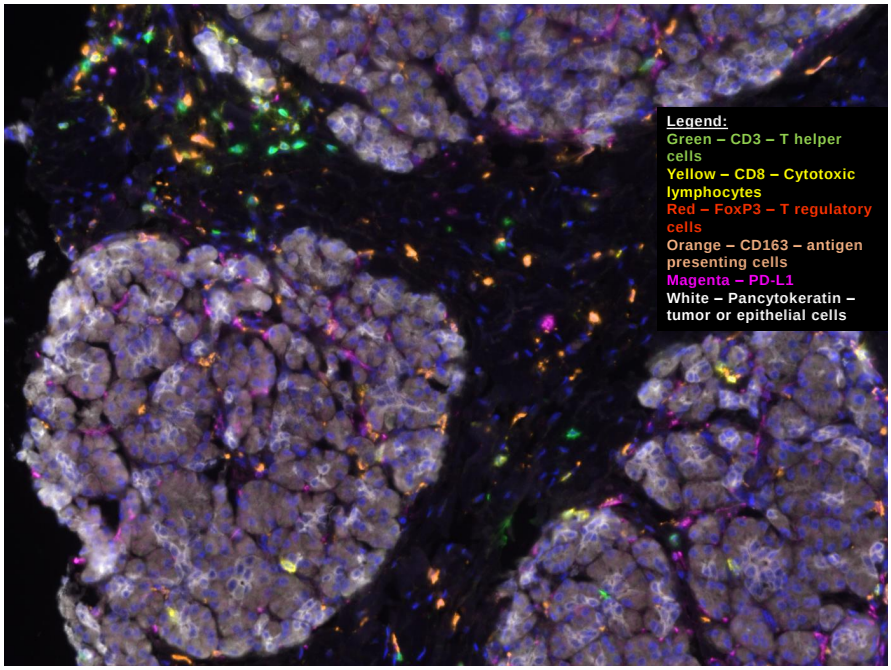


Figure 4.1: A sample multiplexed immunofluorescence (mIF) image, with the different colours signifying the fluorescence corresponding to different surface bio-markers on the cells imaged. Image courtesy Dr. Timothy L. Frankel.

phenotype markers in a mIF image of chronic pancreatitis.

Classifying point sets from mIF images is important because it provides a novel way for pathologists to diagnose diseases. For example, in the context of chronic pancreatitis and pancreatic ductal adenocarcinoma, the point sets from mIF images describe the spatial relationships between the diseases' cells, which reveals information about how the interactions between these cells vary.

This problem is challenging due to the following three reasons. First, the points are distributed non-uniformly in the space, which results in complex spatial relationships. Second, the contributions of different spatial relationships vary between different classification tasks, which requires that the representation of the relationships be adjusted to meet the need of specific tasks. Third, spatial relationships between cells of different types are both crucial and implicit in point sets, and the small number of available learning samples makes it difficult for deep neural networks (DNNs) to learn these spatial relationships without appropriate neural network architectures.

Cell.X.Position	Cell.Y.Position	Treg	APC	Epithelial	HelperT	PDL1_CD3	PDL1_CD8	PDL1_FoxP3	CD4	CTLs
867	12	neg	neg	neg	neg	neg	neg	neg	pos	neg
27	17	neg	neg	neg	neg	neg	neg	neg	pos	neg
1104	20	neg	neg	neg	neg					neg
1214	24	neg	neg	neg						

Location | Attributes (e.g., surface phenotype markers)

Figure 4.2: A point set from a multiplexed pathology image.

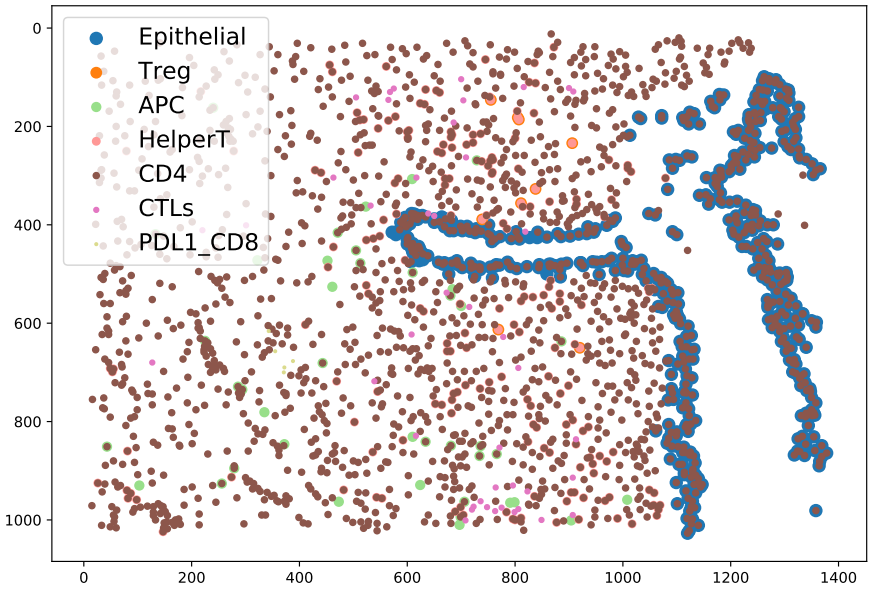


Figure 4.3: A map of a point set from a sample Chronic Pancreatitis mIF image.

Manual morphologic or cell-count based methods, which are the conventional clinical approaches for studying spatial patterns within mIF images, are limited by inter-observer variability. Substantial efforts have been made to apply machine learning techniques to automate the pathology diagnosis process alongside the expansion of digital imaging techniques. In particular, deep neural networks (DNNs) have been extensively studied in a large number of pathology diagnosis applications, including pixel/patch-level region-of-interest detection [76, 77, 78] as well as image-level decision [79, 80] for various diseases, which have shown state-of-the-art results. However, most of the existing DNN-based applications take images as the input and are inapplicable for our problem. The disadvantage of working with raw images is that the variation in staining and artifacts present across all images in a given cohort may influence analysis. In contrast, point sets offer a simplified representation of cell locations and neighbourhoods, invariant of cell borders and cellular morphology. Recently, as point cloud data from LiDAR scanners have become increasingly popular, the representation of point sets has attracted more attention [81]. However, current methods mainly focus on point sets with few numerical attributes, such as signal strength, and they do not handle categorical attributes specifically. Hence, they do not take full advantage of the spatial relationships between points of different categories.

To overcome these limitations, we propose a new DNN architecture, namely SRNet, with novel design of spatial-configuration based representation learning layers. Experiments show that the proposed methods yield much higher accuracy than the competing DNN methods. Our contributions can be summarized as follows:

- We introduce a deep neural network architecture, SRNet, to learn a representation of the spatial relationships between points of different categories that are not captured by the commonly used statistics such as the cross-K function and the participation ratio.

- We conduct rich experimental studies to evaluate the accuracy of the proposed methods. The discovered crucial patterns are verified by domain scientists, confirming the method's potential to help pathologists identify novel spatial relationships between different cell types (e.g., immune cells and tumor cells) in the micro-environment.

Scope: The scope of this study is limited to analyzing point datasets representing the location and types of cells derived from multiplexed immuno-fluorescence (mIF) images to distinguish between diseases. Analyzing mIF images without converting them to point sets is outside the scope of this paper. In addition, we do not evaluate the proposed method with larger datasets due to a lack of public benchmarks. Field trials to evaluate the clinical value of the proposed method also fall outside the scope of this study.

4.2 Problem Definition & Data Description

Given a collection of categorical point sets (e.g., cells with different surface phenotype markers) from multiplexed immuno-fluorescence (mIF) images and the class labels of the point sets (e.g., different diseases), the goal of this study is to train a machine learning model that distinguishes between the point sets of different classes. The primary objective is to achieve a high classification accuracy.

We define a categorical point set as a collection of points, where each individual point belongs to a single category and is located in 2-D Euclidean space. This study was conducted on 199 anonymized point sets derived from mIF images belonging to two disease groups, namely chronic pancreatitis(i.e., class 1) and pancreatic ductal adenocarcinoma (PDAC) (i.e., class 2), which had 56 and 143 point sets, respectively. In the original dataset, cell surface makers indicate nine phenotypes. Each cell might be associated with one or more phenotype. To transform the original point sets into categorical point sets, we considered any point that had a single phenotype marker as belonging to the category corresponding to that phenotype, and we replaced every point that had multiple phenotype markers with a group of points, having one marker each and then assigned the points to multiple categories corresponding to the phenotype of each point's marker. Generating point sets from multiplexed pathology images is beyond the scope of this paper, and we treat point sets as given inputs.

4.3 Related Work

The history of deep neural network (DNN) methods that directly take point sets as the input dates back to PointNet [81], which learns point features independently through multiple fully connected neural network layers and aggregates them into a shape feature using a max pooling layer. These methods have been widely used for 3D shape classification and semantic segmentation as the point clouds collected from LiDAR scanners have become increasingly popular. PointNet++ [82] defines multi-scale regions and uses PointNet to learn their features. It then hierarchically aggregates the regions' features, so it can capture local configurations and learn fine-grained patterns. Similar to PointNet++, the idea of spatially partitioning points and then recursively aggregating them has been extensively explored. For example, KD-trees are used in [83, 84] to spatially partition points based on point density.

Meanwhile, much effort has been made to introduce DNN architectures that were originally designed for other data formats (e.g., imagery and time series). For example, convolutional neural network (CNN) models are studied in the spectral domain (e.g., RGCNN [85]) and the spatial domain (e.g., Pointwise convolution [86]). Recursive neural network (RNN) models are applied with the assumption that "order matters" [87], and there are autoencoders that learn the representation of point sets [88]. However, these models are not specifically designed to handle multi-categorical point sets and do not take full advantage of the spatial relationships between different categories of points.

4.4 Proposed Approaches: SRNet

The cross-category spatial neighborhood relationships is an important component in the spatial configuration of points. In pathology diagnosis, the spatial correlations between different types of immune cells may vary with diseases, which inspires us to introduce a deep neural network (DNN) method, namely spatial-relationship aware neural network (SRNet), with novel representation layers to represent point sets with the spatial relationships between different categories of points in them.

4.4.1 Spatial-Relationship Quantification

An intuitive way of representing the spatial relationships of point sets consisting of various categories is to utilize measures quantifying the relationships. In this subsection, we present two measures for spatial relationships widely used in spatial data mining and spatial statistics, and how they can be used in classification tasks.

Participation ratio

The participation ratio quantifies the degree to which a category tends to be involved in a co-location pattern. Co-location patterns [58, 1] refer to set of categorical point sets that tend to be located in close proximity, such as a point set of Nile crocodiles and Egyptian plovers [89].

A co-location pattern [58] has three defining concepts. First, a co-location pattern is in the form of a set of categories. Second, a neighborhood clique is a set of points within which every pairwise distance is smaller than a threshold. Third, an instance of a spatial co-location pattern is a neighborhood clique composed of one point from every category in the pattern. The participation ratio (PR) of a category in a co-location pattern is then defined as the ratio of the points in the category that are within the instance of the pattern, which is calculated as:

$$PR(c_i, p) = \frac{|c_i \text{ points in the instances of } p|}{|c_i \text{ points}|}, \tag{4.1}$$

where c_i is a category and p is a spatial co-location pattern, and $|\cdot|$ yields the cardinality of a set. The value of a participation ratio is between 0 and 1. The greater the value, the more likely c_i points are located nearby the points of other categories in the pattern p.

For the sake of computational efficiency, in this study we only consider the spatial co-location patterns composed of two categories, so Equation 4.1 can be transformed as:

$$PR(c_i, c_j, d) = \frac{|c_i \text{ points with } c_j \text{ in } SN(ci, d)|}{|c_i \text{ points}|}, \tag{4.2}$$

where $SN(ci, d)$ yields a circular spatial neighborhood with a radius of d around a c_i point. Given a point set containing points belonging to k categories and a neighborhood

distance threshold, there will be $k(k-1)$ participation ratios. An important hyperparameter that affects the value of the participation ratio is the neighborhood distance threshold. Participation ratios with different neighborhood distance thresholds imply the relationships between points in different spatial scales, so we compute the participation ratios with a collection of l neighborhood distance thresholds. Therefore, we can use a vector of $k(k-1)l$ participation ratios as the representation of a point set with k categories.

To validate that the spatial relationships quantified by participation ratios may be useful for distinguishing between the point sets of different diseases, we plot the probability density distribution of four participation ratios in the dataset we described in Section 4.2 using histograms in Figure 4.4. Each histogram has ten equal-width bins that represent the range of participation ratio values, and the area of each bin is the probability density of the bin. As can be seen, the probability distribution of a participation ratio varies with category pairs as well as with neighborhood distance thresholds, and in Figure 4.4a and 4.4c, the probability distributions for the two diseases are notably different. Therefore, $PR(\text{APC}, \text{Treg}, 100)$ and $PR(\text{APC}, \text{Treg}, 200)$ may be used to distinguish the point sets of the two diseases.

Ripley's cross-K function

The participation ratio, $PR(c_i, c_j,\ d)$, can be thought of as the expectation that c_j points exist in the spatial neighborhood c_i point. However, the existence of c_j points does not tell the whole story about the distribution of c_j points in a c_i points' spatial neighborhood. Ripley's cross-K function, instead, focuses on the number of c_j points in c_i points' spatial neighborhood. It is defined in the following form:

$$\text{cross-K}(c_i, c_j, d) = \frac{E(|c_j \text{ in } SN(c_i, d)|)}{E(|c_j \text{ in entire study area}|)}, \tag{4.3}$$

where c_i and c_j are two categories, d is a neighborhood distance threshold, $SN(c_i, d)$ yields the circular spatial neighborhood of a c_i point with a radius of d, and $E(\cdot)$ returns the expectation. The value of a cross-K function is non-negative. The greater the value, the more c_j points are located nearby the c_i points. Similar to how we represent a point set using its participation ratios, given l neighborhood distance thresholds, we can also

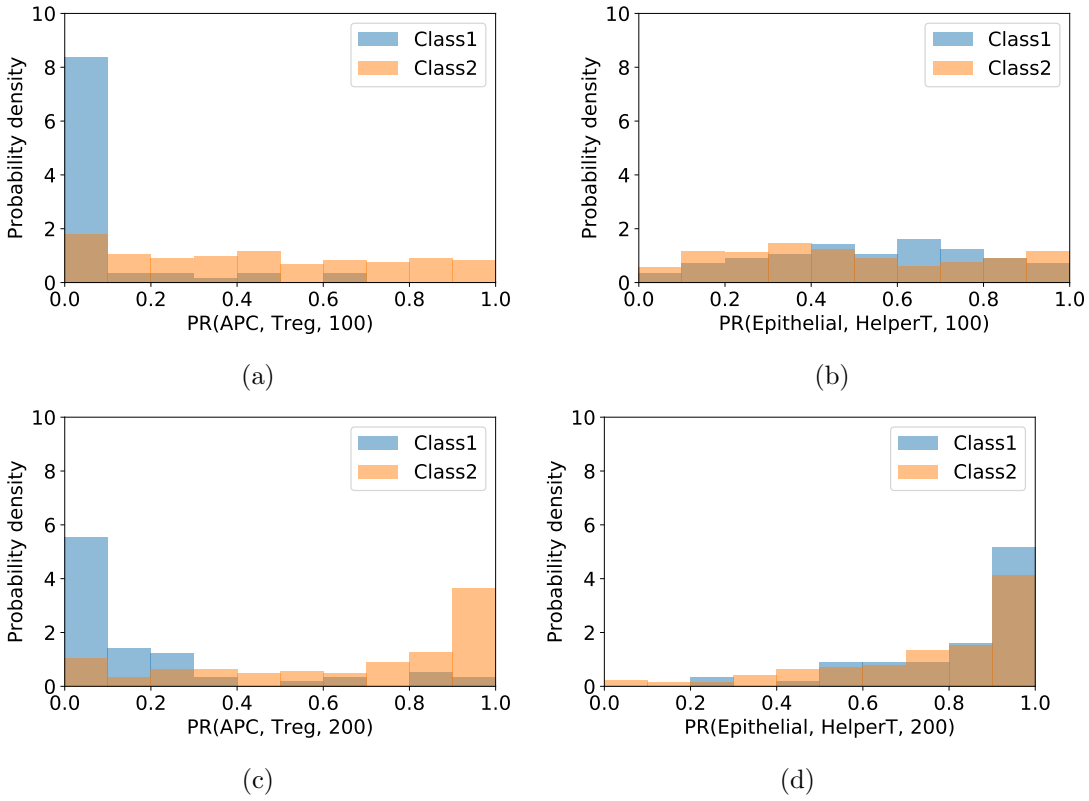


Figure 4.4: Examples of the probability distribution of participation ratios.

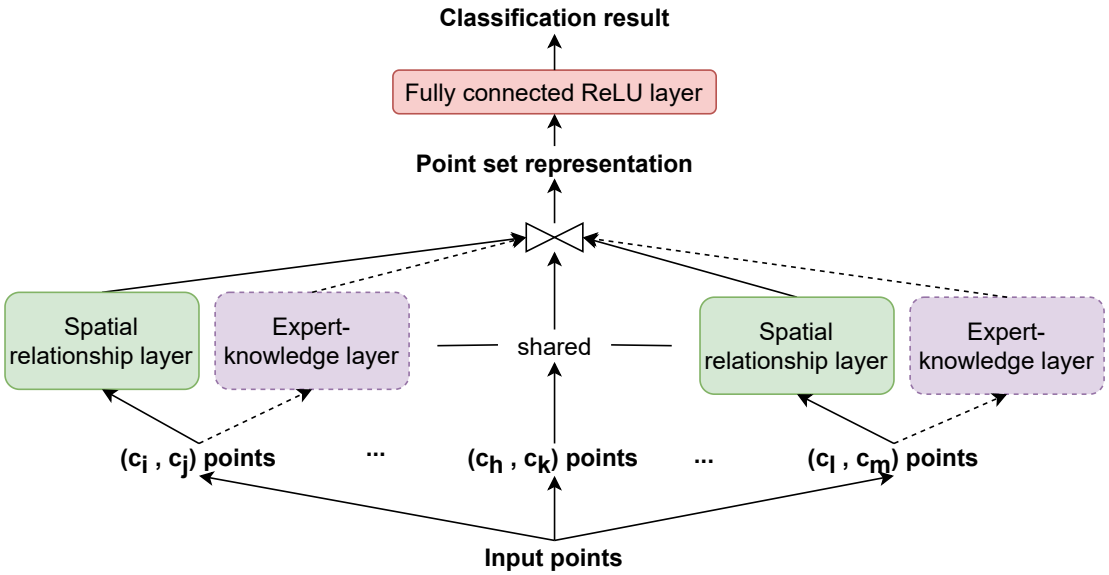


Figure 4.5: Overview of the SRNet architecture.

represent a point set with k categories using a vector contains $k(k-1)l$ cross-K function values.

4.4.2 Proposed SRNet Architecture

In the definitions of the participation ratio and the cross-K function, a core component is the representation of the spatial neighborhood of points. Given an ordered category pair (c_i, c_j), and a spatial neighborhood distance threshold d, the participation ratio uses the existence of c_j points and the cross-k function uses the count of c_j points to represent the distribution of c_j points in the spatial neighborhood of c_i points. However, in addition to existence and count, there may be other patterns that describe the spatial relationships between c_i and c_j points. Hence, we design a DNN model that uses a spatial-relationship aware neural network (SRNet) that learns the spatial distribution of c_j points in c_i points' spatial neighborhood for every ordered category pair (c_i, c_j), and then to generate a representation of point sets. The point-set representation can then be fed into a fully connected neural network for classification.

Figure 4.5 shows the overall architecture of SRNet. The input of the approach is a categorical point set denoted as $X \in \mathbb{R}^{N\times(D+1)}$, where N is the number of points and

$D = 2$ is the spatial dimensions. Each point has one categorical attribute, and there are k categories in total. Similar to using the participation ratios or the cross-K function values to represent point sets, the SRNet uses a DNN layer (spatial relationship layer) to learn the spatial relationship measures of all $k(k-1)$ ordered category pairs. This architecture facilitates the integration of human expert knowledge by concatenating the learned spatial relationship measures with the measures provided by human experts (e.g., the participation ratio, the cross-K function). The architecture of the spatial relationship layer, shown in Figure 4.6, has three main components: a spatial neighborhood layer (Section 4.4.2), a spatial distribution attention layer (Section 4.4.2), and a weighted average pooling layer. For every ordered category pair (c_i, c_j), the spatial neighborhood layer generates a representation of the spatial distribution of c_j points in every c_i point's spatial neighborhood, and the spatial distribution attention layer learns the attention to be paid to each c_i point according to the spatial distribution of c_i points. Then, the weighted average pooling layer aggregates the spatial neighborhood representation of every c_i point with different weights to calculate the spatial relationship measures of pair (c_i, c_j). Finally, the spatial relationship measures of all ordered category pairs are concatenated to generate the overall representation of the point set, denoted as $Y \in \mathbb{R}^{k \times (k-1) \times W}$, where W is the feature dimension of the spatial relationship measures of a category pair.

Spatial neighborhood layer

Given an ordered category pair (c_i, c_j), a spatial neighborhood layer is applied to represent the spatial distribution of c_j points within every individual c_i point's spatial neighborhood independently. The input of this layer is a c_i point and the c_j points in its spatial neighborhood, and its output is a vector representing the spatial distribution of the c_j points. There are two main steps in this layer, namely, spatial location representation and spatial distribution summarization (Figure 4.7).

Spatial location representation focuses on representing the relative location of a c_j point in the spatial neighborhood of a c_i point. The most commonly used representation of a relative location is the difference of coordinates. However, it was reported in [90] that the difference of coordinates failed to convey the information of various spatial distributions. Recently, Gao et al. proposed a representational model that uses the

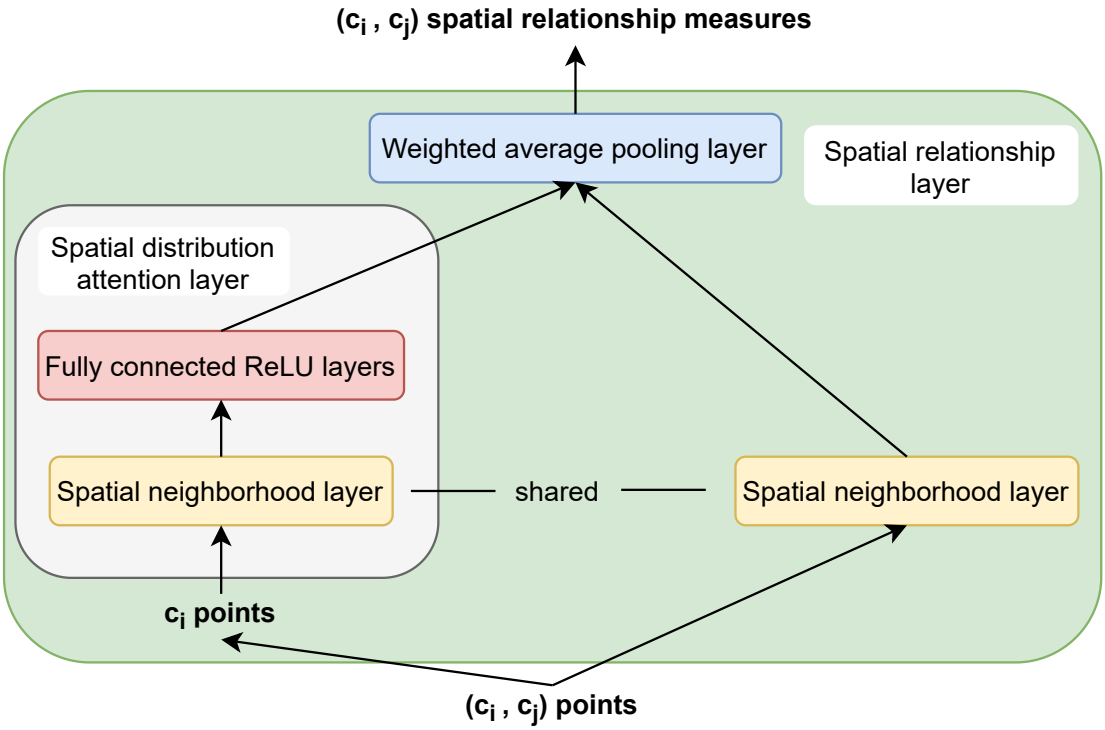


Figure 4.6: The architecture of the spatial relationship layer.

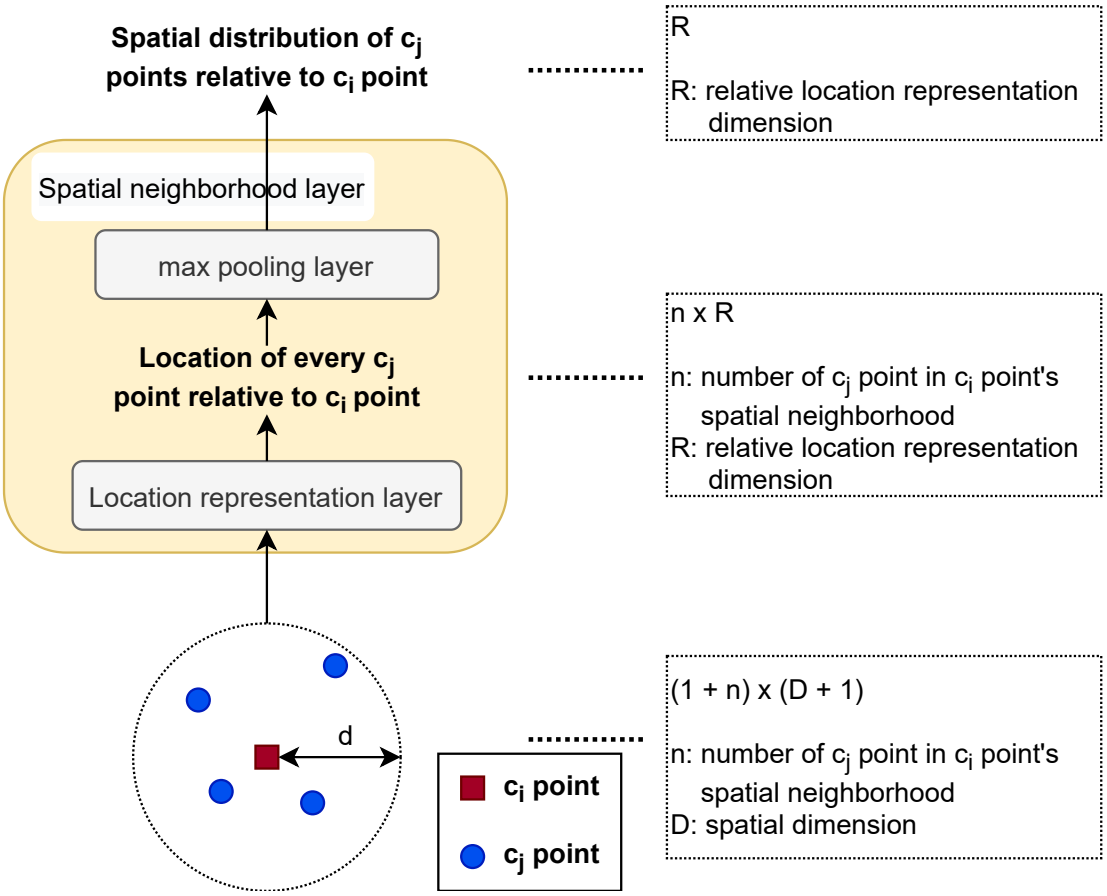


Figure 4.7: The architecture of the spatial neighborhood layer.

hexagon patterns of the grid cells to form a high-dimensional vector representation of 2D locations (x), based on the following theorem whose proof is given in [91].

Theorem 4.4.1. *Let* $\Psi(x) = (e^{i\langle a_j, x\rangle}, j = 1, 2, 3)^T \in \mathbb{C}^3$ *where* $e^{i\theta} = \cos\theta + i\sin\theta$ *and* $\langle a_j, x\rangle$ *is the inner product of* a_j *and* x. $a_1, a_2, a_3 \in \mathbb{R}^2$ *are 2D vectors such that the angle between each pair is* $2\pi/3, \forall j, \|a_j\| = 2\sqrt{\alpha}$. *Let* $C \in \mathbb{C}^{3\times 3}$ *be a random complex matrix such as* $C * C = I$. *Then* $\phi(x) = C\Psi(x)$, $M(\Delta x) = Cdiag(\Psi(\Delta x))C*$ *satisfies*

$$\phi(x + \Delta x) = M(\Delta x)\phi(x) \tag{4.4}$$

and

$$\langle \phi(x + \Delta x), \phi(x)\rangle = d(1 - \alpha\|\Delta x\|^2) \tag{4.5}$$

where $\phi(x)$ *is the representation of location* x, $d = 3$ *is the dimension of* $\phi(x)$, *and* Δx *is a small displacement from* x.

Based on Theorem 4.4.1, Mai et al. [90] introduced a multi-scale location representation model by using sine and cosine functions of different frequencies in $\Psi(x)$, inspired by the multi-scale periodic representation of grid cells in mammals [92]. In this model, $\Psi(x)$ is represented as a concatenation of the position embedding (PE) at S scales, $PE(x) = [PE_1(x); ...; PE_s(x); ...PE_S(x)]$,

$$PE_s(x) = [PE_{s,1}(x); PE_{s,2}(x); PE_{s,3}(x)], \tag{4.6}$$

$$\begin{aligned} PE_{s,j}(x) = [\cos(\frac{\langle x, a_j\rangle}{\lambda_{min} \cdot g^{s/(S-1)}}); \sin(\frac{\langle x, a_j\rangle}{\lambda_{min} \cdot g^{s/(S-1)}})], \\ \forall j = 1, 2, 3, \end{aligned} \tag{4.7}$$

where $a_1 = [1, 0]^T, a_2 = [-1/2, \sqrt{3}/2]^T, a_3 = [-1/2, -\sqrt{3}/2]^T$ are unit vectors, the angles between every pair of vectors is $2\pi/3$, $\lambda_{min}, \lambda_{max}$ are the minimum and maximum grid scales, and $g = \frac{\lambda_{max}}{\lambda_{min}}$. The matrix multiplication $C\Psi(x)$ is represented as $NN(PE(x))$, where $NN(\cdot)$ are fully connected ReLU layers. Therefore, the location of a c_j point relative to a c_i point can be represented as $NN(PE(\Delta x))$, where Δx is the difference of their coordinates.

Given a collection of relative location representations of c_j points in a c_i point's spatial neighborhood, a max pooling layer is applied to summarize the relative locations to get the representation of the c_i point's spatial neighborhood. Pointnet[81] has theoretically and experimentally demonstrated that with enough neurons, a max pooling layer is able to learn to summarize a point distribution [81].

Spatial distribution attention layer

To get the representation of the spatial relationship measures of pair (c_i, c_j), an average pooling layer is used to aggregate the representation of c_j points' distribution in all the spatial neighborhoods of c_i points. However, it is questionable whether all c_i points should contribute equally to the spatial relationship measures. In their study of the spatial co-location patterns, Barua and Sander discovered that the spatial distribution of the points belonging to a category within a co-location pattern affected the statistical significance of the pattern's participation ratio where all points contributed equally [93]. A potential reason is the existence of spatial auto-correlation. In other words, the spatial neighborhoods of nearby points are similar. If all points contribute equally, the spatial neighborhood of a point away from other points may be overwhelmed by the spatial neighborhoods of the points in clusters. Therefore, we introduce a spatial distribution attention layer to determine the attention paid to each c_i point when generating the spatial relationship measures of (c_i, c_j). The layer first generates the representation of the spatial distribution of c_i points in each c_i point's spatial neighborhood independently using the proposed spatial neighborhood layer. Then it estimates the attention paid to each c_i point according to the representations using multiple fully connected ReLU layers. This method is similar to the application of farthest point sampling (FPS) in PointNet++ [82], which selects subsets of representative points to learn local features. Instead of using a greedy heuristic as in FPS, the proposed spatial distribution attention layer uses neural network layers to adjust the attention to points.

4.5 Experiment

Our experimental evaluation has two components: (1) a comparison of the proposed methods with the state-of-the-art deep neural network (DNN) point set classification

methods; and (2) an analysis of the importance of the spatial relationship measures.

4.5.1 Classification Accuracy Comparison

We have conducted two sets of experiments: (1) comparing our proposed methods: handcrafted features using classic spatial measure (i.e., participation ratio or cross-k function) and learned features using SRNet, each combined with a simple neural network classifier, with the state-of-the-art (SOTA) DNN point set classification methods (i.e., PointNet and PointNet++), (2) comparing handcrafted features combined with simple classification models with the SOTA DNN point set classification methods. The experiments are designed to answer the following questions: 1) did the proposed method yield more accurate classification results than the competing DNN methods? 2) how do the spatial relationship measures used to represent point sets affect classification accuracy? 3) how does the choice of classification method affect accuracy? Classification accuracy is measured by AUC-ROC, precision, recall, F1 score, and accuracy. The candidate methods compared were as follows.

- **PointNet**[81]: PointNet is a neural network architecture that directly consumes point sets for applications ranging from object classification to part segmentation.
- **PointNet++**[82]: PointNet++ is a hierarchical neural network architecture that applies PointNet recursively to capture local structure and recognize fine-grained patterns and complex scenes.
- **PR + DT / RF/ NN**: The point set representation composed of the participation ratios (Section 4.4.1) is fed into a decision tree / random forest / fully connected neural network model for classification.
- **cross-K + DT / RF/ NN**: The point set representation composed of the cross-K function values (Section 4.4.1) is fed into a decision tree / random forest / fully connected neural network model for classification.
- **SRNet / +PR / +cross-K**: The point set representation learned by the SRNet model proposed in Section **??** without human expert knowledge / with the participation ratio measures / with the cross-K function measures is fed into a fully connected neural network model for classification.

Table 4.1: Classification accuracy results.

Method	**AUC-ROC**	**Precision**	**Recall**	**F1 score**	**Accuracy**
PointNet	0.518 (0.026)	0.352 (0.079)	0.518 (0.026)	0.421 (0.120)	0.508 (0.160)
PointNet++	0.529 (0.089)	0.412 (0.138)	0.529 (0.089)	0.421 (0.138)	0.529 (0.089)
PR+DT	0.903 (0.027)	0.955 (0.028)	0.911 (0.036)	0.932 (0.016)	0.905 (0.021)
PR+RF	0.979 (0.011)	0.936 (0.025)	0.949 (0.027)	0.942 (0.022)	0.917 (0.031)
PR+NN	0.980 (0.016)	0.948 (0.035)	0.954 (0.041)	0.950 (0.025)	0.929 (0.035)
cross-K+DT	0.852 (0.011)	0.911 (0.027)	0.914 (0.058)	0.911 (0.027)	0.874 (0.031)
cross-K+RF	0.955 (0.028)	0.852 (0.019)	**0.967** (0.017)	0.906 (0.015)	0.856 (0.023)
cross-K+NN	0.938 (0.027)	0.908 (0.037)	0.933 (0.046)	0.919 (0.025)	0.883 (0.036)
SRNet	0.939 (0.030)	0.951 (0.038)	0.884 (0.066)	0.914 (0.031)	0.884 (0.039)
SRNet+PR	**0.985** (0.015)	**0.967** (0.002)	0.962 (0.040)	**0.964** (0.020)	**0.950** (0.014)
SRNet+cross-K	0.964 (0.022)	0.953 (0.028)	0.909 (0.047)	0.930 (0.028)	0.904 (0.037)

The implementation of both PointNet and PointNet++ are available on GitHub [1]. The decision tree, the random forest, and the fully connected neural network methods were implemented using the Python scikit-learn package [94]. The maximal depth of the decision tree methods was set to 4, and the maximal depth and the number of estimators of the random forest methods were set to 3 and 1000. Other hyperparameters were kept as the default values. The fully connected neural network classifier had two hidden ReLU layers with 4096 neurons and a sigmoid layer as the output layer.

The SRNet method was implemented using PyTorch, where the spatial neighborhood of each point was set as a circle with a radius of 200, and the minimal grid cell size, the maximal grid cell size, and the number of scales of the multi-scale location representation layers were set at 1, 100, and 10 respectively. All the spatial neighborhood layers shared the same architecture and parameters. The fully connected ReLU layers in the spatial neighborhood layers had four hidden layers, and the hidden layer dimension was set at 256. The feature dimension of the learned spatial relationship measures of each ordered category pair was 32. The SRNet and the neural network classifier were trained using the Adam optimization algorithm with the learning rate of 10^{-4} to minimize the cross entropy loss of the classification results and the ground truth.

[1] Link to PointNet repository: `https://github.com/charlesq34/pointnet`. Link to PointNet++ repository: `https://github.com/charlesq34/pointnet2`.

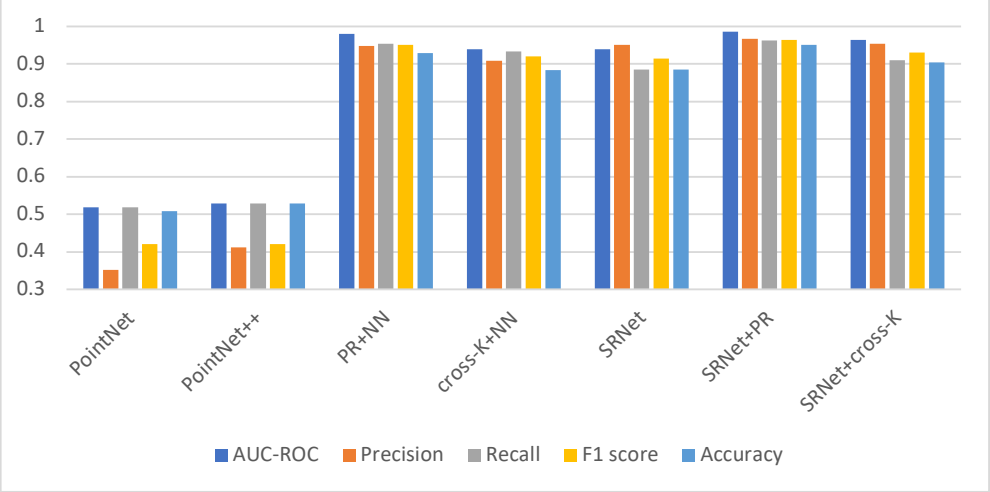


Figure 4.8: The classification accuracy of the methods using neural network classifiers.

We used the dataset described in Section 4.2. Since the original dataset only had 199 point sets, we used 5-fold cross-validation and augmented the number of point sets by partitioning, flipping, and rotating the original point sets. To get subsets of a point set and keep spatial relationship information in each subset, instead of randomly sampling points, we partitioned the minimum bounding rectangle (MBR) of the point set horizontally by 20% and 80% and then 80% and 20%, and used the 80% subsets. The subsets were then flipped both horizontally and vertically. Finally, the flipped subsets were rotated by 90 degrees three times. Thus, after data augmentation, there were $199 \times 2 \times 4 \times 4 = 6368$ point sets in total.

Table 4.1 shows the mean and standard deviation (in parentheses) of classification accuracy measures of the candidate methods. The highest accuracy is highlighted in bold. It is evident that the proposed methods, even a very simple model (e.g., the decision tree model) with a well-defined spatial relationship measures (e.g., the participation ratio), were much more accurate than the competing DNN point set classification methods (i.e., PointNet and PointNet++).

A comparison of the classification accuracy of the methods using neural network classifiers (Figure 4.8), shows that the methods using classic spatial relationship measures (PR+NN and cross-K+NN) and those using measures learned by the proposed SRNet (SRNet, SRNet+PR, SRNet+cross-K) had much higher accuracy than the competing

DNN methods. This indicates that the proposed SRNet was able to learn spatial relationship measures that were missed by the competing DNN methods. Moreover, the accuracy of the SRNet+PR and SRNet+cross-K methods was higher than that of the PR+NN and cross-K+NN methods, respectively, which means the proposed SRNet is able to learn features that are not captured by the participation ratio and the cross-K function but that were useful for the classification task.

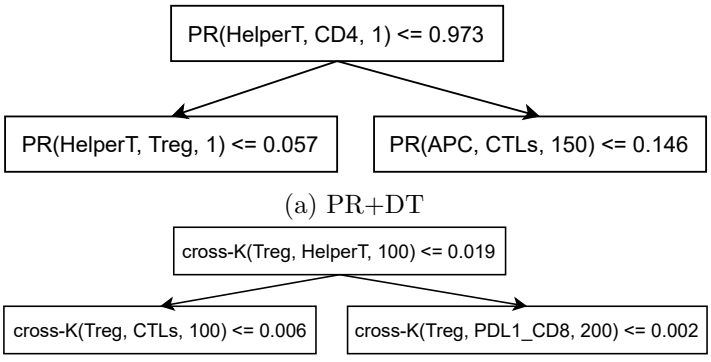
Finally, the classification accuracy of methods using the same point set representation (e.g., PR+DT v.s. PR+NN) indicates that complex models yielded more accurate results. However, the effect of choosing different classification methods on classification accuracy was not as significant as the effect of point set representation.

4.5.2 Analysis of Spatial Relationship Measures

The goal of the second set of experiments was to analyze the category pairs whose spatial relationship measures are important for classifying the point sets of the two diseases, as this provided a way to discover the interactions between cells that varied with diseases.

In the PR+DT and cross-K+DT methods, the feature vectors composed of the participation ratios and cross-K function values were fed into decision tree models. Since in every node of the decision tree model, a feature is selected greedily to divide samples into two groups according to a heuristic (e.g., the information gain), the selected features indicate which category pairs contain high variation in their spatial relationships. Figure 4.9 shows the first two layers of the decision trees trained using the entire dataset described in Section 4.2. As can be seen, the spatial relationships between HelperT cells and CD4 cells and between Treg cells and HelperT cells were significantly different under the micro environment of the two diseases.

In the PR+RF and cross-K+RF methods, the feature vectors composed of the participation ratios and cross-K function values were fed into random forest models. Feature importance in the random forest models can be measured by the mean impurity decrease, which also implies the spatial relationships between the category pairs vary a lot in the point sets of the two diseases. Table 4.2 lists the top ten important features in the PR+RF and cross-K+RF models trained using the entire dataset. As can be seen, both the participation ratio features and cross-K function features indicate that the spatial relationships between the HelperT and Treg cells are most useful for distinguishing the



(a) PR+DT

(b) cross-K+DT

Figure 4.9: First two layers of the decision trees trained using the entire dataset in Section 4.2

point sets of the two diseases.

For the PR+NN, cross-K+NN, and the SRNet methods, we evaluated the importance of the spatial relationship measures, namely, the participation ratio, the cross-K function value, and the representation learned by SRNet, through permutation feature importance. Permutation feature importance measures the increase in the prediction error of the model after we permute the feature's values. In this experiment, the importance of the spatial relationship measures of an ordered category pair was measured by the classification accuracy after exchanging the corresponding elements in the representation vectors. The lower the accuracy, the more important the measures of ordered category pair. In the dataset described in Section 4.2 the most important ordered category pairs were (HelperT, Treg), (HelperT, CD4), (CTLs, Treg), and (APC, Treg).

4.5.3 Clinical Implications

From a clinical perspective, the results highlight some key cell phenotype relationships that may directly or indirectly play a role in the disease micro-environment. Specifically, the relationship between CTLs and T-regs, and Helper T-cells and T-regs are of particular interest from an immunological standpoint.Cytotoxic Lymphocytes(CTLs) are the cells that actively seek out and kill cancer cells in the environment on activation of the immune system[72]. On the other hand, under normal conditions, the T-regulatory cells have a regulating effect on the immune response of the locale [71]. It has been observed

Table 4.2: Top 10 important features obtained in the PR+RF and cross-K+RF methods.

Rank	**PR+RF feature**	**cross-K+RF feature**
1	PR(HelperT, Treg, 1)	cross-K(Treg, HelperT, 100)
2	PR(HelperT, CD4, 1)	cross-K(HelperT, Treg, 200)
3	PR(HelperT, Treg, 50)	cross-K(HelperT, Treg, 50)
4	PR(HelperT, Treg, 200)	cross-K(HelperT, Treg, 100)
5	PR(HelperT, Treg, 100)	cross-K(HelperT, Treg, 1)
6	PR(HelperT, Treg, 150)	cross-K(Treg, HelperT, 50)
7	PR(CD4, Treg, 150)	cross-K(Treg, HelperT, 1)
8	PR(CD4, Treg, 200)	cross-K(HelperT, Treg, 150)
9	PR(CD4, Treg, 100)	cross-K(Treg, HelperT, 200)
10	PR(APC, Treg, 100)	cross-K(Treg, HelperT, 150)

that T-regulatory cells play a more functional role in the cancer micro-environment, and there is potential for some interplay between the two cell phenotypes from a functional standpoint. Due to this, there is a tendency for them to co-localize at a higher frequency with CTLs, and potentially inhibit their function [73]. This may be due to physiologic suppression of activated CTLs, or pathological polarization of CD4 positive cells by tumor secreted factors in the tumor micro-environment[74]. Further investigation on a larger cohort to confirm the potential discriminatory power of the pairwise interactions observed in this experiment would be warranted.

The identification of the cell-pairs opens up a potential for a novel method to capture the difference in cellular arrangements across different diseases. This also alludes to the influence of cell-cell distances and their relative placement in the state of the micro-environment [32]. Along with reinforcing known relationships, these features would also serve to offer new insight into potential cell-cell relationships that were either unknown or little explored in previous studies. In the age of increasing focus on personalized treatment paradigms, the utilization of a spatially-aware approach would assist physicians in making more informed treatment plans.

4.6 Conclusion & Future works

In this chapter, we proposed a deep learning point-set classification method, namely SRNet, for multiplexed pathology images. SRNet provides a novel way for pathologists to diagnose diseases. Instead of classifying multiplexed immuno-fluorescence (mIF) images directly, we first converted mIF images to point sets representing the cells on mIF images, and then classified the point sets. An experimental evaluation showed that the proposed SRNet can learn spatial relationship measures that are not captured by classic measures, and the classification accuracy of using the learned measures significantly outperformed the SOTA deep learning point-set classification methods, reaching 95% accuracy (about 80% more accurate). In addition, the proposed methods helped to discover pairs of cell types that might inspire new pathology findings.

In the future, we will compare the proposed method on point sets with the methods directly analyzing mIF images without converting them to point sets. We also plan to identify larger mIF images and other spatial pathology datasets for larger and broader evaluation of the proposed method. In addition, the proposed SRNet focuses on the spatial relationships between two cell types, and we plan to extend its capability by taking the relationships between multiple cell types into consideration.

Chapter 5

Physics-guided Energy-efficient Path Selection Using On-board Diagnostics Data

5.1 Introduction

Given a spatial graph, two nodes in the graph as the origin and destination, and historical on-board diagnostics (OBD) data, the energy-efficient path selection (EPS) problem aims to find the most energy-efficient path between the origin and the destination. Figure 5.1 illustrates a sample input of the EPS problem consisting of a spatial graph with eleven nodes (i.e., $n1, n2, ..., n11$) and twelve edges (i.e., $e1, e2, ..., e12$), six traces of OBD data (i.e., $t1, t2, ..., t6$), and two nodes $n1$ and $n5$ as the origin and the destination. Figure 5.2 shows six traces in the OBD data in Figure 5.1. Every trace is in the form of an ordered sequence of records, and each record is composed of an edge and the status of the vehicle on the edge. Energy consumption and average speed are two examples of the status attributes. Suppose that the expected energy consumption of the vehicle on path $[e1, e5, e8, e11, e12, e10, e7, e4]$ is the lowest among those on all possible paths linking $n1$ and $n5$, the path is the energy-efficient path for this example.

Monitoring and managing traffic and transportation systems using the OBD data collected from telematics devices on connected vehicles is an important component of

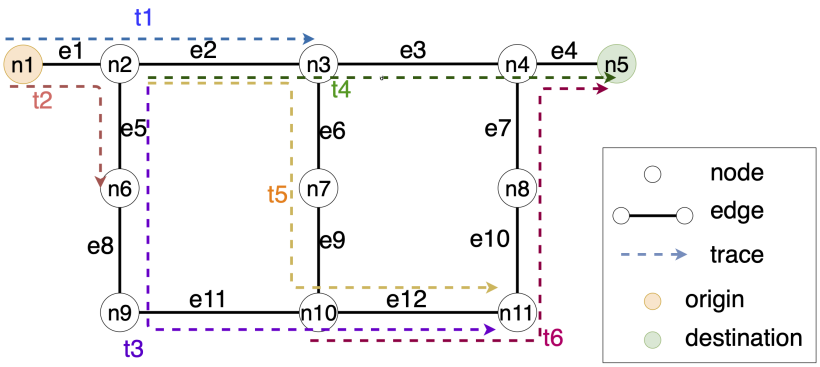


Figure 5.1: A spatial graph with six traces of on-board diagnostics data.

$t1$

edge	Energy consumption	Average speed
$e1$	2	9
$e2$	9	30

$t2$

edge	Energy consumption	Average speed
$e1$	1	5
$e5$	1	6

$t3$

edge	Energy consumption	Average speed
$e5$	3	8
$e8$	2	5
$e11$	4	10
$e12$	3	9

$t4$

edge	Energy consumption	Average speed
$e2$	7	15
$e3$	9	35
$e4$	2	7

$t5$

edge	Energy consumption	Average speed
$e2$	9	40
$e6$	2	30
$e9$	2	20
$e12$	7	30

$t6$

edge	Energy consumption	Average speed
$e12$	3	15
$e10$	2	10
$e7$	2	8
$e4$	2	4

Figure 5.2: Sample OBD data with six traces.

a smart city. This chapter describes our work on leveraging the OBD data to provide energy-efficient routing suggestions. Two main objectives of the vision for smart cities are sustainability and prosperity, both of which benefit from the ability to estimate and reduce the energy consumption for transportation. Reports show that car exhaust emissions contribute significantly to air pollution and anthropogenic climate forcing [22], and energy consumption for transportation in the United States cost more than $507 billion in 2015 [23]. Despite the efforts to reduce energy consumption for transportation, such as electric car research supported by the U.S. Department of Energy [95], the expected energy use continues to climb [96].

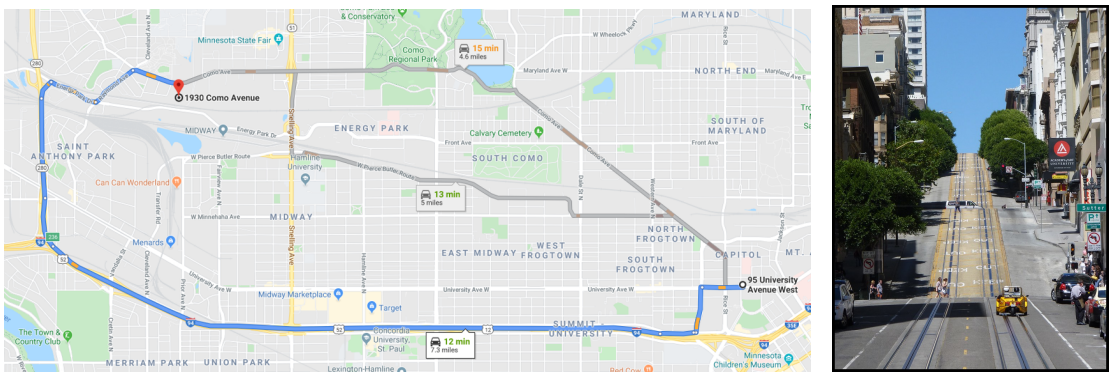
Geo-referenced OBD data facilitate accurate travel cost estimation and novel path selection algorithms using the estimation. Previous research on energy-efficient path selection has demonstrated that the potential energy saving is about 10% by taking energy-efficient routes instead of fastest routes [97, 98]. A McKinsey Digital report also estimates that personal geo-referenced data could help save consumers about $600 billion by 2020, by providing routes to vehicles that avoid traffic congestion through next-generation routing algorithms [99]. Equation 5.1 is a simplified powertrain energy consumption model commonly used in mechanical engineering, where the meaning of the symbols are in Table 5.1 [100]. Briefly, the energy consumption of a vehicle is determined by the vehicle's motion properties (i.e., t, a, v, and v_h) as well as its physical parameters (i.e., m, A, c_{air}, and η). Between two places there are often multiple possible paths, and different paths have have different spatiotemporal features such as speed limit, traffic and road conditions, which affect the motion properties of a vehicle and its energy consumption in turn. For example, Figure 5.3a shows that between two places there are a fast but long path through highways and two slow but short paths through local roads. The high speed on highways reduces the time cost, but may make the thermal energy due to air resistance, which is represented by the term with v^3 in Equation 5.1, dominate in energy consumption. The road test we conducted in Cincinnati, OH (detailed in Section 5.7) indicated that the expected energy consumption on the energy-efficient path composed of local roads is about 38% lower than that on the fastest path composed of highways. In addition, the existence of up/down hill roads (e.g., in San Francisco (Figure 5.3b)) also affects vehicles' energy consumption. Therefore, in this chapter we propose a method to leverage OBD data for energy-efficient path selection.

Table 5.1: Physics model symbols

Symbol	Physical Interpretation
W	work (energy consumption)
η	vehicle's powertrain system efficiency
m	vehicle's mass
A	vehicle's front surface area
c_{air}	air resistance coefficient
c_{rr}	rolling resistance coefficient
a	acceleration
v	velocity
v_h	vertical velocity
t	time
g	gravity acceleration
ρ	air density

$$W = \frac{1}{\eta}[\int (mav)dt + \int (mc_{rr}gv)dt + \int (\frac{A}{2}c_{air}\rho v^3)dt + \int mgv_h dt]. \tag{5.1}$$

The EPS problem, which is a variant of the shortest path selection problem, has two sub-tasks, namely, the prediction of expected energy consumption (EEC) of a path and the selection of the most energy-efficient path. The challenges of predicting EEC of a path are two-fold. The first challenge is the dependence of EEC on physical parameters of vehicles, which is different from the cost metrics (i.e., distance, time) for the shortest or fastest path selection problems. For example, a construction truck consumes more energy than a sedan when traveling along a path following the same velocity profile, even though the distance and the time cost are the same. Moreover, the autocorrelation of the energy consumption on different segments of a path prevents edge-centric travel cost estimation models from estimating accurately. In other words, the EEC of a path is a property of the entire path, but not the sum of the EEC of individual edges along the path. The selection of an energy-efficient path given an EEC estimation model also has two challenges. The first is the high computational cost of EEC estimation which is needed for every candidate path in currently existing path selection algorithms. The second challenge is that the EEC on a path may be negative because of regenerative braking,



(a) Highway vs. local roads. (b) Up/down hill roads.

Figure 5.3: There are often multiple paths between two places.

which makes popular path selection algorithms like Dijkstra's algorithm inapplicable, since they assume that travel cost is non-negative.

Most of the methods for predicting EEC are composed of two steps [101, 102, 103, 104, 98]: 1) predict or obtain the velocity profile along the objective path, and 2) estimate EEC using the velocity profile via a vehicle simulation model. However, accurate velocity profile prediction is challenging since a large number of factors may affect it, some of which (e.g., the schedule of vehicles and pedestrians) are unpredictable in real-world road systems. A novel EEC estimation model using OBD data without velocity profile is needed. In addition, the commonly-used path selection algorithms are based on Dijkstra's or the Bellman-Ford algorithm [105, 106], and apply a "path + edge" pattern to explore candidate paths. These algorithms evaluate the travel cost of a candidate path once the path is explored. However, the estimation of travel cost is computationally expensive, and some estimation is redundant in cases where the OBD data on an edge is a subset of the OBD data on a path containing the edge. Therefore, a novel path selection algorithm is needed which adopts a "path + another path" pattern when exploring candidate paths.

Our preliminary work [12] and [11] proposed a physics-guided energy consumption (PEC) model for the prediction of EEC and a maximal-frequented-path-graph shortest-path (MFPG-SP) algorithm using the PEC model for the path selection.

This chapter extends our previous work by proposing two algorithms for the sub-task of path selection given an EEC estimation model, analyzing their correctness and

completeness, and validating the algorithms through experiments and two case studies.

The contributions in the chapter are as follows:

1. We propose an A*-like admissible heuristic and an informed maximal-frequented-path-graph shortest-path (IN-MFPG-SP) algorithm.

2. We propose a maximal-frequented-path-graph label-correcting (MFPG-LC) algorithm that can handle negative energy consumption in a maximal frequented path graph.

3. We analyze the proposed algorithms for correctness, completeness, and computational time complexity.

4. We evaluate the proposed algorithms via controlled experiments with real-world and synthetic data.

5. We conduct two case studies using the OBD data collected from three UPS trucks in 18 months to evaluate the potential energy saving of adopting the proposed method compared to using the path recorded in the data, and to show that the proposed method can suggest paths that are more energy-efficient than the paths suggested by the commonly-used path selection tools.

6. We conduct one road test using one UPS truck to validate that the proposed method can suggest paths that are more energy-efficient than the paths suggested by the commonly-used path selection tools.

Scope: The goal of this chapter is to introduce a method of suggesting more energy-efficient paths according to our estimation, compared with the previously used paths and the fastest paths suggested by the currently available tools. Current initiatives such as USDOE ARPA-E NEXTCAR [107] and EU optiTruck [108] consider path selection to be an importance approach to reduce energy consumption by vehicles. Previous research has also shown that path selection can save up to 10% of energy [97, 98], and it can complement other energy saving methods related to the refinements of vehicle characteristics, driving behaviors, road surface, traffic flow, etc. Identifying the main factors to energy consumption is out of the scope of this chapter .

5.2 Basic Concepts and Problem Definition

In this section, we introduce the basic concepts in this study, based on which the energy-efficient path selection problem is formally defined.

5.2.1 Basic Concepts

A **spatial graph** $G_s = (N_s, E_s)$ consists of a set of **spatial-node**s N_s and a set of **spatial-edge**s E_s, where each element $n \in N_s$ is a geo-referenced point, and each element $e = (n_i, n_j) \in E_s$ is an edge that joins spatial-node n_i and spatial-node n_j. Figure 5.1 shows an example of a spatial graph where circles represent nodes (e.g. $n1$, $n2$) and lines represent edges (e.g. $e1$, $e2$). A road system is an example of a spatial graph where nodes are road intersections and edges are road segments.

A **path** in a graph is a sequence of edges linking an ordered sequence of nodes. The first and the last nodes are the origin and the destination of the path respectively. Given two paths ϕ_1 and ϕ_2, ϕ_1 is a **sub-path** of ϕ_2 if the edges of ϕ_1 are all along ϕ_2. In Figure 5.1, path $[e1, e2]$ is a sub-path of path $[e1, e2, e3]$. The union ($\cup$) of two paths $\phi_1 \cup \phi_2$ at a node shared by them is composed of the edges of ϕ_1 before the node and those of ϕ_2 after the node. For example, in Figure 5.1, $[e2, e3, e4] \cup [e3, e7]$ at $n4$ is $[e2, e3, e7]$.

A **trace** in an on-board diagnostics (OBD) dataset is a map-matched multi-attributed trajectory of a vehicle. Each trace has a list of records. Each record in a trace is composed of an edge and a set of vehicle status on the edge, such as, the total energy consumption, the average speed, and the state of charge. Figure 5.1 shows six traces of OBD data as dashed arrows whose detailed records are in Figure 5.2. For example, trace $t4$ has three records, indicating that the vehicle travels along edges $e2, e3, e4$ and its energy consumption and average speed on these edges are 7,9,2 and 15,35,7 respectively.

A **frequented path** (FP) is a path along which there are at least a certain number of traces of OBD data in the same direction. A **union of frequented paths** (UFP) is a path composed of a union of two or more FPs such that it is not the sub-path of any other FP. The FPs are paths where we have most historical OBD data, and UFPs are formed by FPs, so we assume the energy consumption estimation on them would be more accurate compared to that on other paths. In addition, FPs and UPFs also imply people's traveling preference. If we set the minimum number of traces of OBD

data along an FP as 1 in Figure 5.1, path $[e1, e2]$ is an FP along which there is a trace $t1$. A sample of UFP is $[e1, e2, e3, e4]$, which is formed by the union of two FPs $[e1, e2]$ and $[e2, e3, e4]$ at $n3$.

Expected energy consumption (EEC) of a path in a spatial graph is the amount of energy expected to be consumed by a vehicle traveling along the path. For the sake of simplicity, in the examples in the chapter , the EEC of an FP is calculated as the average energy consumption of the traces along it. For example, in Figure 5.1 there is only one trace $t1$ on the entire path $[e1, e2]$, so the EEC of each edge along path $[e1, e2]$ is $[2, 9]$ according to $t1$, and the EEC of the path is $2 + 9 = 11$. The EEC of an UFP is the summation of the EEC of the FP forming it, and the EEC of the overlapping edges of multiple FPs is the average EEC of the FPs on the edges. For example, in Figure 5.1 the UFP $[e1, e2, e3, e4]$ is formed by two FPs $[e1, e2]$ and $[e2, e3, e4]$, and the two FPs overlap on $e2$. Since the EEC of the edges on $[e1, e2]$ is $[2, 9]$ and that on $[e2, e3, e4]$ is $[7, 9, 2]$, the EEC of the edges on $[e1, e2, e3, e4]$ is $[2, 8, 9, 2]$, and that of the entire path is $2 + 8 + 9 + 2 = 21$. This method of estimating EEC is path-centric, because only traces along the entire path are considered when estimating the EEC of an FP. For example, when estimating the EEC of path $[e1, e2]$, the path-centric methods only use $t1$, while the traditional edge-centric methods will first estimate the EEC on $e1$ according to $t1$ and $t2$ as well as the EEC on $e2$ according to $t1, t4$, and $t5$. However, part of the energy consumption of $t2, t4$ and $t5$ is for factors other than traveling on the path, such as the right turn of $t2$ at node $n2$ and the start of $t4$ on edge $e2$.

An **energy-efficient path** between two locations is the path with the least EEC according to our estimation using historical OBD data among all possible paths between the locations. For example, the energy-efficient path between $n1$ and $n5$ is the path $[e1, e5, e8, e11, e12, e10, e7, e4]$ whose EEC is 18, since the other two possible paths $[e1, e2, e3, e4]$ and $[e1, e2, e6, e9, e12, e10, e7, e4]$ have a total cost of 21 and 25 respectively.

In our preliminary work [11], we defined two other key terms for the maximal-frequented-path-graph shortest-path algorithm.

A **maximal frequented path** (MFP) is an FP that is not the sub-path of any other FP. Since any sub-path of an FP is an FP, only estimating the EEC of MFPs eliminates the redundant computation of estimating the EEC of an FP and its sub-paths repeatedly.

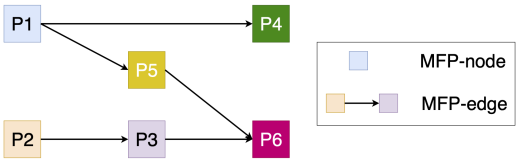


Figure 5.4: The MFPG for the spatial graph and OBD data in Figure 5.1 (The color of each MFP matches the trace on it).

A **maximal frequented path graph** (MFPG) of a spatial graph is a directed graph representing the MFPs and the relationship between each pair of them. Thus, the nodes of an MFPG (**MFP-node**s) are MFPs. An edge of an MFPG (**MFP-edge**) exists between two MFP-nodes (ϕ_1 and ϕ_2) if $\phi_1 \cup \phi_2$ form an UFP at a spatial-node. Suppose that the minimum number of OBD traces along an FP is 1. Each OBD trace in Figure 5.1 defines an MFP. The spatial graph and the OBD data in Figure 5.1 can be transformed into an MFPG, as shown in Figure 5.4. Each square is an MFP-node representing an MFP in the spatial graph, while the directed arrows are the MFP-edges between them. MFP-node Pi is the MFP defined by the trace ti. For example, $P1$ is the path $[e1, e2]$ where $t1$ is along.

We say a path in a spatial graph is represented by a path in a MFPG, if the path in the spatial graph is a sub-path of the UFP formed by the MFP-nodes along the path in the MFPG, and the origin and the destination of the path in the spatial graph are on the first and the last MFP-nodes respectively. Since an MFP contains several spatial-nodes in it, multiple paths in a spatial graph could be represented by a path in an MFPG. For example, the MFPG-nodes in path $[P5, P6]$ in the MFPG form an UFP $[e2, e6, e9, e12, e10, e7, e4]$ in the spatial graph, so UFPs such as $[e2, e6, e9, e12, e10]$, $[e2, e6, e9, e12, e10, e7]$, and $[e6, e9, e12, e10]$ are represented by it. The cost of a path in an MFPG is defined as the cost of the UFP made up of all MFP-nodes in the path and excluding the spatial-edges before the origin. For example, let $n3$ be the origin, the cost of path $[P5, P6]$ in the MFPG is the cost of UPF $[e6, e9, e12, e10, e7, e4]$, which is $[e6, e9, e12] \cup [e12, e10, e7, e4]$, in the spatial graph. We prove that, given a spatial graph and its MFPG, every FP is a sub-path of at least one MFP, and every UFP can be formed by the union of at least one collection of MFPs. Therefore, we can model the FPs and UFPs in a spatial graph without losing any information via an MFPG.

5.2.2 Problem Definition

We formally define the energy-efficient path selection problem as follows:

Input:

- A spatial graph.
- Historical OBD data of vehicles in the graph.
- A minimum threshold for the number of OBD traces along an FP.
- Two spatial nodes o and d.

Output: An energy-efficient path between o and d.

Objective: Avoid a path that is energy-inefficient.

Constraints: The resulting path is an FP or an UFP.

An example of the problem we are solving in this chapter is in the following form: We are given the spatial graph shown in Figure 5.1, six traces on it with details shown in Figure 5.2, the minimum number of traces along an FP as 1, and two spatial nodes $n1$ and $n5$ as the origin and destination. The output of the problem would be the path $[e1, e5, e8, e11, e12, e10, e7, e4]$ with a total cost of 18. In this chapter , we only focus on finding energy-efficient paths among FPs and UFPs. Energy consumption estimation on paths without enough OBD data is outside the scope of this chapter .

5.3 Related Work and Preliminary Results

5.3.1 Related Work

Based on the basic spatial unit where travel cost is estimated, the existing path selection methods can be categorized into two groups, i.e., edge-centric, and path-centric methods (left branch in Figure 5.5).

Edge-centric methods assume the travel cost on individual edges is independent, and the travel cost of a path is the sum of the costs on the edges along the path. Dijkstra's [105] and the Bellman-Ford [106] algorithms are widely applied with an assumption that the cost of traveling on each edge is a constant. Other studies based on them have focused on accelerating computation [109, 110, 111, 112] or introducing new constraints [113, 114], new cost metrics [115, 116], and new cost representation [117, 118, 119, 120]. Most of the currently available energy-efficient path selection methods belong to this

group [97, 98], which suggest path with the lowest energy consumption according to the estimated energy consumption on individual edges. However, all the edge-centric methods suffer from the fact they ignore the dependence between the costs of different parts along a path.

Rather than thinking of a path as a sequence of individual edges, path-centric methods treat it as a sequence of overlapping sub-paths[121, 12]. Since the basic unit to estimate the cost of a path is a sub-path, these path-centric methods maintain the dependence between the costs of different parts along a path. This is beneficial to energy consumption estimation. For example, Figure 5.6 shows a road intersection b on a highway from a to d, where there is an entrance ramp from c. There are two OBD traces ($t1$ and $t2$). $t1$ is along the highway, while $t2$ is from the entrance to the highway at b. The energy consumption of the traces on each edge is annotated. To estimate the expected energy consumption (EEC) of the path from a to b then to d, an example of edge-centric solutions is to sum up the average energy consumption on edges a to b and b to d individually. The average energy consumption on edge a to b is 3 kWh according to trace $t1$, and that on edge b to d is 2 kWh according to traces $t1$ and $t2$. Thus, the EEC of the path from a to b then to d is $3 + 2 = 5$ kWh according to the edge-centric solution. Instead, if applying a path-centric solution that only uses traces along the whole path (i.e., $t1$), we will get the result of 4 kWh. Intuitively, a part of the energy consumed by a vehicle after entering a highway is for acceleration, which should not be included in the EEC of a vehicle traveling on a highway. If we use the traces that do not lie along the whole path (e.g., $t2$ in this case), we may mistakenly include energy consumption caused by factors not on the path (e.g., the traffic light at the entrance ramp). Therefore, our path selection algorithm uses a path-centric travel cost estimation model.

Despite their advantages of using path-centric cost estimation model, the PACE and Physics-guided methods in [121, 12] require a lot of redundant computation. The general framework for a path selection algorithm is shown in Algorithm **??**. Given a spatial graph, an origin, and a destination, as well as a travel cost estimation model, the algorithm generates a path satisfying certain criteria. The main steps of the algorithm are as follows. A set of candidate paths CP is initialized in Line 1, typically using the paths consisting of one edge from the origin. Then in each iteration (Lines 2-8), the most promising path in CP is extended, and the result path is added to CP. The

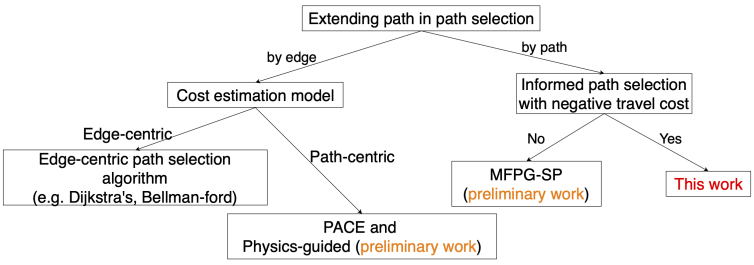


Figure 5.5: A tree of related works.

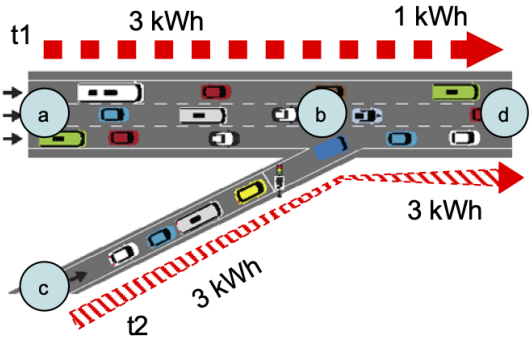


Figure 5.6: Difference between edge-centric and path-centric view at a highway ramp.

iteration ends when the stop criterion is met. The related work implements these steps in different ways. For example, the Dijkstra algorithm's stop criterion is that a path is found between the origin and the destination, while the most promising path in CP is the one with the smallest cost.

All the existing path selection algorithms explore the candidate paths in Line 4 following the pattern of "path + edge", including the PACE and Physics-guided methods. In other words, a candidate path is generated by adding an edge to the end of an old path. These algorithms estimate the EEC of each candidate path after it is explored, which results in redundant computation. Take Figure 5.1 as an example. Exploring candidate paths at $n6$ given the current path $[e1, e5]$, a path $[e1, e5, e8]$ would be the candidate path following the pattern of "path + edge". However, according to the OBD data, all traces passing $n9$ are on the same path from $n2$ to $n11$, which means the estimated cost of the path $[e5, e8]$ would be the same as the corresponding part of the path $[e5, e8, e11, e12]$. The travel cost of the path $[e5, e8]$ is estimated repeatedly when the paths $[e5, e8]$, $[e5, e8, e11]$, and $[e5, e8, e11, e12]$ are explored. Therefore, in our preliminary work, we proposed an algorithm which applies a "path + another path" pattern [11].

Algorithm 4 General path selection algorithm framework

Require:
G: A spatial graph;
o and d: Two nodes;
model: A cost estimation model.
Ensure: The path between o and d satisfying the criteria.
1: candidate paths $CP \leftarrow$ initialization;
2: **while** stop criteria are not met **do**
3: $p \leftarrow$ the most promising path in CP;
4: **for all** extensions p's of p **do**
5: compute the cost of p';
6: add p' to CP;
7: **end for**
8: **end while**

5.3.2 Preliminary Work

Our preliminary work included a physics-guided energy consumption (physics-guided) model [12], and an algorithm based on a maximal frequented path graph [11].

Scenario-based physics-guided energy consumption model

We proposed two energy consumption models to estimate the expected energy consumption (EEC) of traveling along paths with enough data, namely, a scenario-based model for frequented paths (FPs), and a FP-union model for union of frequented paths (UFPs).

The physics-guided model is a path-centric model based on the simplified powertrain energy consumption model in Equation 5.1. We denote the part of energy used for air resistance as $AIR = \int(\frac{A}{2\eta}c_{air}\rho v^3)dt$, and denote the part of energy used for rolling resistance and acceleration as the product of a vehicle parameter factor $V = \frac{m}{\eta}$ and a motion property factor $M = \int(av + c_{rr}gv)dt$. Equation 5.1 can be written as

$$W = AIR + V \times M, \tag{5.2}$$

where W, AIR and M are vectors whose elements represent their values on each edge, while V is a scalar determined by the vehicle.

We clustered the traces along each FP into k scenarios using the K-means algorithm, a popular clustering method, according to their M, which is determined by motion properties, and AIR, which is determined by motion properties as well as a vehicle's front area and powertrain system efficiency. The traces in each group record vehicles of similar motion properties, front surface area, and powertrain system efficiency, but varied mass due to cargo loads. The energy consumption of the traces in a scenario will be a linear function of V, whose intersect and slope are the traces' shared AIR and M respectively. The detailed initialization, update and assignment steps of the K-means algorithm are discussed in [12].

We also propose a FP-union model based on the path decomposition method introduced in [121] to evaluate the energy consumption of a trace-union path (UFP). The key to estimating energy consumption along a UFP is to join the scenarios of adjacent FPs in the decomposition of the UFP according to the M and AIR of each scenarios on the shared edges of the adjacent FPs. A scenario on a FP is joined with the scenario with

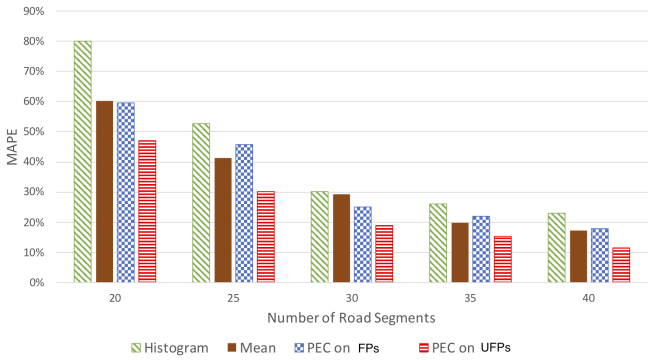


Figure 5.7: MAPE of candidate methods with varying path length.

the most similar M and AIR on the next FP in the path decomposition. The similarity of M and AIR and the method to calculate V are discussed in detail in [12].

We experimentally compared the accuracy of the physics-guided model, including the scenario-based model for FPs and the FP-union model for UFPs, with two statistics which are commonly used as ground truth in state-of-the-art related work, mean of the historical energy consumption, and the distribution of historical energy consumption represented by a histogram. The data used included a road system from OpenStreetMap, and OBD data collected from a vehicle in Fort Worth, TX from 1/1/2017 to 3/1/2018, which contained 10759 traces.

Figure 5.7 shows the mean absolute percentage error (MAPE) of estimates provided by the histogram method, the mean method, and the physics-guided model on FPs, as well as the physics-guided model on UFPs with the minimum number of traces on a FP as 10. As can be seen, all methods became more accurate as the number of edges increases. The physics-guided model was clearly the most accurate on FPs. On UFPs, it achieved similar accuracy as the mean method, even though it requires much less data.

Maximal-Frequented-Path-Graph Shortest-Path (MFPG-SP) Algorithm

The MFPG-SP algorithm explores new candidate paths following a "path + another path" pattern in Line 4 of Algorithm **??** using the maximal frequented path graph (MFPG) of the input spatial graph. Because given a spatial graph, the path to be found in this problem is either an FP or a UFP, and all FPs and UFPs are represented by an MFP-node or a path in the MFPG of the input data, the MFPG-SP algorithm is

composed of two steps: 1) find the MFP-node or the path in the MFPG that links the origin and destination, and 2) get the objective path in the spatial graph represented by the found MFP-node or the path in the MFPG. Since the exploration of candidate paths is conducted in the MFPG, and each MFP-node is an MFP, the paths in the spatial graph represented by the candidate paths are explored following the pattern of "path + another path". In this way, we avoid the redundant computation for estimating the expected energy consumption of the sub-paths of MFPs. The details of the MFPG-SP algorithm following the general framework Algorithm **??** are as follows. In Line 1, the set of candidate paths CP is initialized with the MFPs where the origin lies. The stop criterion in Line 2 is that there is no candidate paths with cost lower than that of the found path from the origin to the destination. The most promising path in CP is the path with the lowest cost. In Line 4, as candidate paths are searched, one MFP is added to the currently most promising path if the two can form a UFP. Once a path is extended to the destination, we estimate its cost and remove it from the candidate path set. If the estimated cost is lower than the current lowest cost, the result path and the lowest cost are updated. The cost estimation method used in Line 5 is provided as an input.

We experimentally compare the performance of the MFPG-SP algorithm against the physics-guided algorithm in [12] on a real dataset, containing 10129 traces of OBD data collected from three UPS trucks in Fort Worth, Texas 1/1/2017 - 6/30/2018. Each trace logs the status of a truck when it moves between two delivery stops. The road system is from OpenStreetMap. The origin-destination (OD) pairs of each energy-efficient path query in the experiments were the OD pairs of the traces of the OBD data. Figure 5.8 shows the results when the the minimum number of traces on a FP is 20. As can be seen, the MFPG-SP algorithm always has a smaller time cost than the physics-guided algorithm. Furthermore, the gap increases and indeed becomes overwhelming with increasing result path length.

Even though the MFPG-SP algorithm realizes "path + another path" pattern when exploring candidate paths, it applies an uninformed search strategy to find the objective path, which does not make any use of the information we have about the destination to help in the search process. An example of the information is the direction of the destination. If the destination is to the east of the origin, a path heading east may be more likely to be the energy-efficient path compared with a path heading west. The

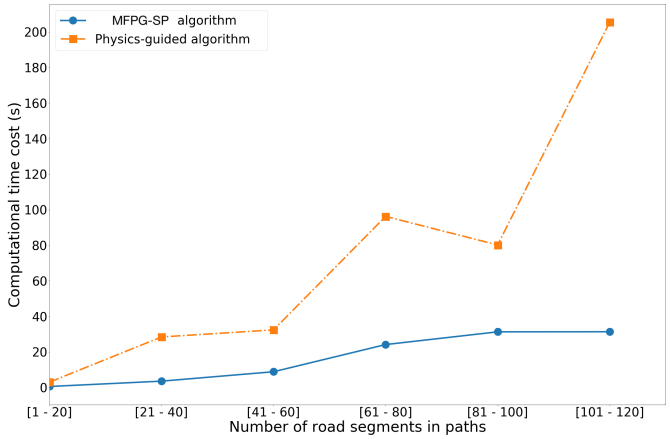


Figure 5.8: Computational time cost of the physics-guided and the MFPG-SP algorithm with varying number of edges in the objective paths.

ignorance of the information may result in redundant computation to explore paths that are impossible to reach the destination. Furthermore, the MFPG-SP algorithm assume travel cost is always positive, which is not applicable for vehicles with regenerative breaking.

5.4 Proposed Approaches

We design an admissible heuristic in a maximal frequented path graph (MFPG) that can guide the search space towards the destination and still guarantee finding the correct path. Then, we apply it in an informed maximal-frequented-path-graph shortest-path (IN-MFPG-SP) algorithm. In addition, we propose a maximal-frequented-path-graph label-correcting (MFPG-LC) algorithm to handle possibly negative energy consumption in the energy-efficient path selection problem.

5.4.1 MFPG Heuristic

Since an MFPG is a representation of a spatial graph and the OBD data on it, we take an admissible heuristic in the spatial graph as an input to compute the heuristic in the MFPG, which we call MFPG heuristic. We name the spatial-nodes on an MFP where the MFP form a UFP with other MFPs as the transfer spatial-nodes of the MFP. Then,

we define the heuristic travel cost from an MFP-node P to the destination as:

$$H(P) = \min_{n \in N_{P(transfer)}} \{h(n)\}, \tag{5.3}$$

where $N_{P(transfer)}$ is the set of transfer spatial-nodes of an MFPG-node P, and $h(n)$ is an admissible heuristic travel cost from a spatial-node n to the destination. When the destination is on MFP-node P, then $H(P) = 0$. If an MFP-node does not contain any transfer spatial-nodes or the destination, its heuristic travel cost would be ∞, since there would be no path from it to the destination.

To use the MFPG heuristic in the EPS problem, we propose an admissible heuristic for the energy consumption of a vehicle from a spatial-node to the destination in a spatial graph based on the physics model shown in Equation 5.1. In this model, the energy is consumed for three purposes, namely, accelerating $\frac{m}{\eta}\int(av)dt$, working against rolling resistance $\frac{mc_{rr}g}{\eta}\int(v)dt$, and working against air resistance $\frac{Ac_{air}\rho}{2\eta}\int(v^3)dt$. A heuristic travel cost is admissible if it always underestimates the actual travel cost in a path selection algorithm, so we find the lower bound of the energy consumption as an admissible heuristic. Suppose that the physical parameters of the vehicle are given. The minimal energy for acceleration is reached when the vehicle keeps a constant speed, in which case the energy for acceleration is 0. The energy for working against rolling resistance is linearly correlated to the travel distance, so its minimum is reached when the distance is equal to the Euclidean distance between the spatial-node and the destination. The energy for working against air resistance increases with the velocity of the vehicle, so its minimum is reached when the velocity is the minimum speed by law in the road system. To ensure that this heuristic is admissible, we make use of the lowest values of c_{rr}, c_{air}, and ρ from the literature and assume the powertrain system η to be 1. Because this heuristic for energy consumption is admissible, we can get the MFPG heuristic for energy consumption in a MFPG by using this heuristic as $h(n)$.

5.4.2 Informed MFPG-SP (IN-MFPG-SP) Algorithm

We adjust the MFPG-SP algorithm by including the MFPG heuristic in the cost of a path in an MFPG to develop the informed maximal-frequented-path-graph shortest-path (IN-MFPG-SP) algorithm. Similar to the MFPG-SP algorithm, the IN-MFPG-SP

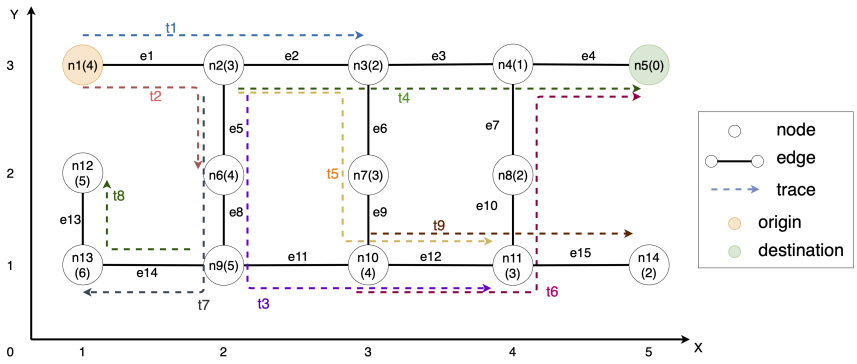


Figure 5.9: Sample data in Figure 5.1 with additional spatial-nodes, spatial-edges, and traces of OBD data.

*t*7

edge	Energy consumption	Average speed
*e*5	1	6
*e*8	1	5
*e*14	1	7

*t*8

edge	Energy consumption	Average speed
*e*14	1	5
*e*13	1	6

*t*9

edge	Energy consumption	Average speed
*e*12	1	8
*e*15	1	7

Figure 5.10: Three new traces of OBD data in Figure 5.9 in addition to those in Figure 5.2.

algorithm has two steps as well. The second step is the same in both algorithms, but the first step of the IN-MFPG-SP algorithm uses the A* algorithm to efficiently find the MFP-nodes or the paths in an MFPG linking the origin and destination. That is to say, in the first step, the IN-MFPG-SP algorithm orders the candidate paths in the MFPG not by the actual travel cost of the path ($C(path)$), but by the full travel cost $F(path) = C(path) + H(P)$, where P is the last MFP of the path, and $H(P)$ is the heuristic travel cost from P to the destination.

To show how the MFPG heuristic helps guide the search space towards the destination we modify the spatial graph from Figure 5.1 to include three more traces, namely, $t7$, $t8$, and $t9$ (Figure 5.9), whose details are shown in Figure 5.10. Suppose we set the minimum number of traces along an FP as 1. All of the added traces lie along new MFPs, since none is along a sub-path of any other FP. The data in Figure 5.9 can be represented by the MFPG shown in Figure 5.11. For simplicity, we assume that the energy consumption

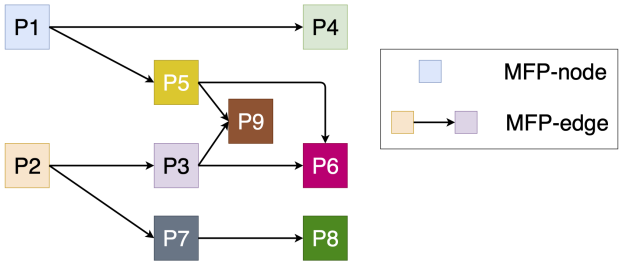


Figure 5.11: The MFPG for the spatial graph and OBD data in Figure 5.9 (The color of each MFP matches the trace on it).

Table 5.2: Heuristic travel cost of MFP-nodes in Figure 5.11.

MFP-node	**Transfer Spatial-nodes**	$\boldsymbol{H(\circ)}$
P1	n3	2
P2	n6	4
P3	n11	3
P4	-	0
P5	n11	3
P6	-	0
P7	n13	6
P8	-	∞
P9	-	∞

is only determined by travel distance, and that the edge length in the spatial graph is 1. We find that the least energy consumption per unit distance is 1 in the OBD data, so we use 1 times the Manhattan distance from a spatial-node to the destination as the admissible heuristic of the spatial graph. The heuristic energy consumption of each spatial-node to the destination $n5$ is annotated in parentheses (Figure 5.9). We calculate the heuristic travel cost of all the MFP-nodes according to Equation 5.3 (Table 5.2). For example, $n3$ is the only transfer spatial-node on $P1$ ($[e1, e2]$) whose heuristic travel cost is 2, so the heuristic travel cost of $P1$ is 2. The destination is on $P4$ and $P6$, so their heuristic travel cost is 0. Since there is no transfer spatial-node on $P8$ and $P9$, their heuristic travel cost is ∞.

First, we refer to the execution trace in Table 5.3 to show how the original MFPG-SP algorithm solves the problem with the new trace data shown in Figure 5.9. In step 1,

Table 5.3: Execution trace of the MFPG-SP Algorithm.

Step	CP	$C(\circ)$	p	**Result Path (Cost)**
1	[P1],[P2]	0,0	[P1]	
2	[P1, P4], [P1, P5],[P2]	10,11,0	[P2]	[P1, P4] (21)
3	[P1, P5], [P2, P3], [P2, P7]	11,3,2	[P2, P7]	[P1, P4] (21)
4	[P1, P5], [P2, P3], [P2, P7, P8]	11,3,4	[P2, P3]	[P1, P4] (21)
5	[P1, P5], [P2, P3, P6], [P2, P3, P9], [P2, P7, P8]	11,12,11,4	[P2, P7, P8]	[P2, P3, P6] (18)
6	[P1, P5], [P2, P3, P9]	11,11	[P2, P3, P9]	[P2, P3, P6] (18)
7	[P1, P5]	11	[P1, P5]	[P2, P3, P6] (18)
8	[P1, P5, P6], [P1, P5, P9]	20,19		[P2, P3, P6] (18)

the set of candidate paths is initialized to include $[P1]$ and $[P2]$, the two MFPs where the origin lies. Since both paths have the same cost we can extend either one. $[P1]$ is extended, which results in $[P1, P4]$ and $[P1, P5]$. $[P1, P4]$ already reaches the origin, so the cost along it is estimated to be 21, and it is removed from the set of candidate paths. At every step the process continues extending the most promising paths one MFP at a time, so in step 2 $[P2]$ is extended, etc. It takes eight steps for the algorithm to finally find the path.

The execution trace of the proposed IN-MFPG-SP algorithm is shown in Table 5.4. Again, step 1 initializes the set of candidate paths to include $[P1]$ and $[P2]$. In this case, however, it is $[P1]$ that should be extended, since its full travel cost ($F(\circ)$) is lower than that of $[P2]$. Sometimes the rank of the candidate paths according to their $F(\circ)$ is different from that according to their $C(\circ)$, when the MFPG heuristic works. For example, in step 3, the original MFPG-SP algorithm extends $[P2, P7]$ while the informed IN-MFPG-SP algorithm extends $[P2, P3]$, since $[P2, P3]$ heads to the direction of the destination. It takes the IN-MFPG-SP algorithm only six steps to find the path. Therefore, the MFPG heuristic can reduce the computational cost.

Table 5.4: Execution trace of the Informed MFPG-SP Algorithm.

Step	***CP***	***C*(∘)**	***H*(∘)**	***F*(∘)**	***p***	**Result Path(Cost)**
1	[P1], [P2]	0,0	2,4	2,4	[P1]	
2	[P1,P4], [P1,P5], [P2]	10, 11, 0	0, 3, 4	10, 14, 4	[P2]	[P1,P4](21)
3	[P1,P5], [P2,P3], [P2,P7]	11, 3, 2	3, 3, 6	14, 6, 8	[P2,P3]	[P1,P4](21)
4	[P1,P5], [P2,P3,P6], [P2,P3,P9], [P2,P7]	11, 12, 11, 2	3, 0, ∞, 6	14, 12, ∞, 8	[P2,P7]	[P2,P3,P6](18)
5	[P1,P5], [P2,P3,P9], [P2,P7,P8]	11, 11, 4	3, ∞, ∞	14, ∞, ∞	[P1,P5]	[P2,P3,P6](18)
6	[P1,P5,P6], [P1,P5,P9], [P2,P3,P9], [P2,P7,P8]	20, 19, 11, 14	0, ∞, ∞, ∞	20, ∞, ∞, ∞		[P2,P3,P6](18)

Analysis of the IN-MFPG-SP algorithm

Because the A* algorithm is complete and correct if the heuristic used is admissible, and the first step of the IN-MFPG-SP algorithm uses it to find an MFP-node or a path in a MFPG representing the energy-efficient path, if the MFPG heuristic is admissible, the IN-MFPG-SP algorithm is complete and correct.

Lemma 5.4.1. *If $h(n)$ is admissible heuristic for a spatial-node n, $H(P)$ is an admissible heuristic for MFP-node P, where $H(P)$ is defined in Equation 5.3.*

Proof. We prove this lemma by contradiction. Assume that $H(P)$ is not admissible when $h(n)$ is admissible. Let $\hat{n}$ be the spatial-node with the smallest actual travel cost to the destination on P, so $\hat{n}$ must be a transfer spatial-node of P. We denote the actual and heuristic travel cost from $\hat{n}$ to the destination as $C(\hat{n})$ and $h(\hat{n})$ respectively. Because $h(n)$ is admissible, $C(\hat{n}) > h(\hat{n})$. Since $H(P)$ is not admissible, the actual travel cost from at least one spatial-node in P to the destination is less than $H(P)$. Then, $H(P) > C(\hat{n})$. Since we have proved that $C(\hat{n}) > h(\hat{n})$, $H(P) > h(\hat{n})$, which results in a contradiction with the definition of $H(P)$. □

The worst-case complexity of the IN-MFPG-SP Algorithm is the same as that of the MFPG-SP Algorithm, and is $O(|E_{MFP}||V_{MFP}|)$, where E_{MFP} and V_{MFP} are the sets of MFP-nodes and MFP-edges respectively.

5.4.3 Maximal-Frequented-Path-Graph Label-Correcting (MFPG-LC) Algorithm

As regenerative braking installed on growing more vehicles, the cases where the expected energy consumption is negative on an edge or a path becomes common, especially along downhill slopes in mountainous areas. To enable the algorithm to deal with edges or paths with negative costs, we propose an algorithm using a label correcting strategy, called MFPG-LC algorithm. This approach does not assume that the first path that reaches an MFP-node is the path with the lowest cost to the MFP-node from the origin. To account for this, a visited node is allowed to be extended. Moreover, the MFPG-SP algorithm terminates once all the candidate paths have a cost greater than the minimum cost of the path found. This would not work if travel cost could be negative, since a candidate path may eventually lead to the destination, via paths that have negative costs and have a cost lower than the current minimum. Therefore, the termination condition need to be adjusted for the negative cost.

To overcome the issue of negative travel cost, we use a technique as follows. Before the first step of the MFPG-SP algorithm, that is, finding the MFP-node or the path in the MFPG linking the origin and destination, all the spatial-edges that have negative costs are identified. The sum of the smallest costs on all these edges indicates the maximum cost that could be regained in the graph. This is called the maximal regain bound. Now, if the costs of all the candidate paths are greater than the cost of the current result path minus the maximal regain bound, it would be impossible for any of these paths to be extended, and eventually reach the destination with a cost lower than the currently found path. Only in this case would the algorithm terminate.

To illustrate the advantage of MFPG-LC algorithm, we modify some of the traces from Figure 5.2 to include negative travel costs and then compare how the MFPG-LG and the original MFPG-SP algorithms handle this case. The modified OBD data is shown in Figure 5.12. The MFPG for the data would stay unchanged as shown in Figure 5.4.

$t1$

edge	Energy consumption
$e1$	2
$e2$	3

$t2$

edge	Energy consumption
$e1$	3
$e5$	5

$t3$

edge	Energy consumption
$e5$	5
$e8$	2
$e11$	4
$e12$	-3

$t4$

edge	Energy consumption
$e2$	1
$e3$	2
$e4$	1

$t5$

edge	Energy consumption
$e2$	9
$e6$	2
$e9$	2
$e12$	-5

$t6$

edge	Energy consumption
$e12$	-3
$e10$	-2
$e7$	-2
$e4$	1

Figure 5.12: Sample OBD Data with Negative Edge Costs.

Table 5.5: Execution trace of the MFPG-SP Algorithm with Negative Edge Costs

Step	***CP***	**Cost**	***p***	**Result Path (Cost)**
1	[P1], [P2]	0,0	[P1]	
2	[P1, P4], [P1, P5], [P2]	4,8,0	[P2]	[P1, P4] (7)
3	[P1, P5], [P2, P3]	8,8	[P2, P3]	[P1, P4] (7)

First, the execution trace of the MFPG-SP algorithm is shown in Table 5.5. The algorithm terminates in step 3 since the cost of the found result path is lower than those of both candidate paths. The path with the lowest cost is $[P1, P4]$ whose cost is 7.

The MFPG-LC algorithm solves this problem as follows. The algorithm estimates the maximal regain bound in the graph first. It identifies all the spatial-edges with negative costs, namely, $e7$, $e10$, and $e12$. Then, it calculates the sum of the minimal costs on these edges, which is -9. Table 5.6 shows the execution trace of the MFPG-LC algorithm. In step 2, the algorithm finds that $[P1, P4]$ already reaches the destination, so it removes the path from the set of candidate paths and estimates the cost of the objective path in the spatial graph. Since the sum of the costs of the candidate paths

Table 5.6: Execution trace of the MFPG-LC Algorithm with Negative Edge Costs

Step	***CP***	**Cost**	***p***	**Result Path (Cost)**
1	[P1], [P2]	0,0	[P1]	
2	[P1, P4], [P1, P5], [P2]	4,8,0	[P2]	[P1, P4] (7)
3	[P1, P5], [P2, P3]	8,8	[P1, P5]	[P1, P4] (7)
4	[P1, P5, P6], [P2, P3]	10,8	[P2, P3]	[P1, P5, P6] (5)
5	[P2, P3, P6]	11	-	[P1, P5, P6] (5)

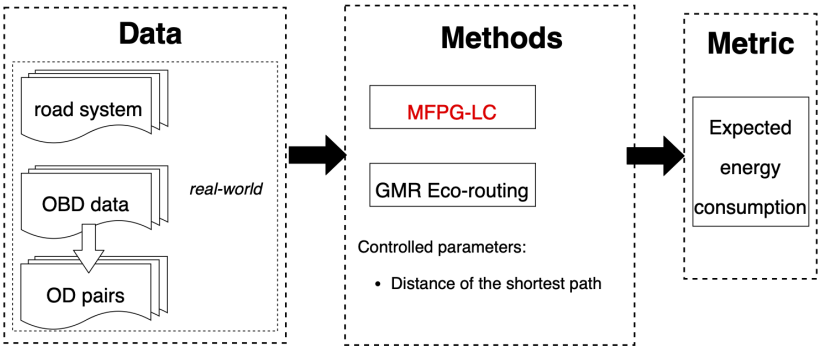
are less than the cost of the current result path minus the maximal regain bound, the algorithm continues to execute. The algorithm terminates in step 5 when there is no candidate path left to explore.

The result path found by the MFPG-LC algorithm linking the origin and destination in the MFPG is $[P1, P5, P6]$, which corresponds $[e1, e2, e6, e9, e12, e10, e7, e4]$. Its expected energy consumption is 5, which is lower than that of the path found by the MFPG-SP algorithm, so the MFPG-SP algorithm terminates before finding the correct result. Hence using the MFPG-LC algorithm would help us find the correct results when travel cost may be negative.

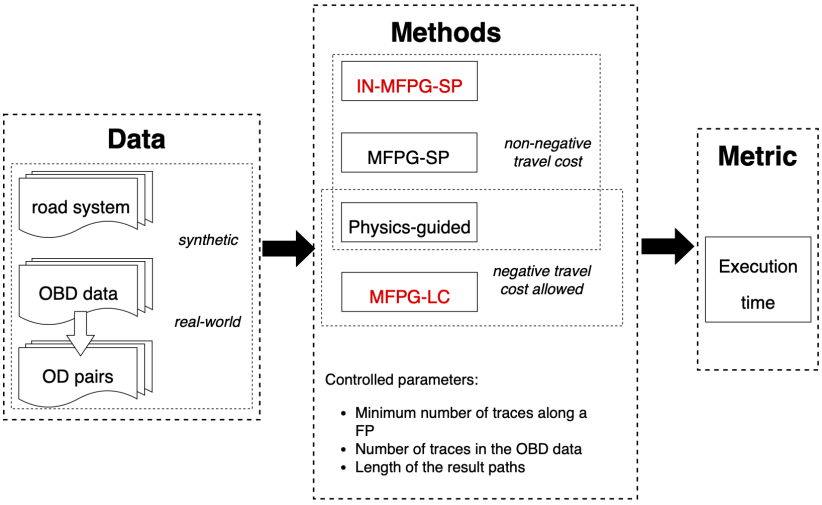
The complexity of the MFPG-LC algorithm is the same as that of the MFPG-SP algorithm, that is, $O(|E_{MFP}||V_{MFP}|)$, where E_{MFP} and V_{MFP} are the sets of MFP-nodes and MFP-edges respectively.

5.5 Experiments

We conducted experiments: 1) to compare the expected energy consumption of the paths suggested by the proposed method (MFPG-LC) and the state-of-the-art energy-efficient path selection method (GMR Eco-routing) [98], and 2) to compare the computational performance of the proposed IN-MFPG-SP and MFPG-LC algorithms against the physics-guided [12] and MFPG-SP [11] algorithms introduced in our preliminary work.



(a) Energy consumption saving experiments



(b) Computational performance experiments

Figure 5.13: Experiment Design.

5.5.1 Experiment Settings

We designed the control experiments shown in Figure 5.13. In energy consumption saving experiments the candidate methods were the proposed method (MFPG-LC) and the state-of-the-art energy-efficient path selection method (GMR Eco-routing) [98]. The GMR Eco-routing method is an edge-centric method, which estimates the expected energy consumption (EEC) on individual road segments, and selects paths according to the estimates. This method estimates EEC by categorizing road segments based on their speed limit, and fits a Gaussian mixture regression model in each category for EEC according to the average speed, speed change, average elevation change, road segment length, and speed limit. In the computational performance experiments, the candidate algorithms for the case where the travel cost was non-negative were the physics-guided [12], the MFPG-SP [11], and the IN-MFPG-SP algorithms. The ones for the case where the travel cost could be negative were the physics-guided [12], and the MFPG-LC algorithms. The metric measuring energy consumption saving was expected energy consumption, which was estimated using the physics-guided energy consumption model, and the metric for computational performance was execution time.

The experiments were designed to answer the following questions:

- Comparative analysis:
  - Is the EEC of the paths selected by the proposed method lower than that of the paths selected by the state-of-the-art methods.
  - Are the proposed algorithms more computationally efficient than the algorithms in preliminary work?
- Sensitivity analysis:
  - How are the proposed methods affected by the number of input traces?
  - How are the proposed methods affected by the minimum number of traces along a FP ?
  - How are the proposed methods affected by the length of the result path ?

The real-world dataset used in the experiments was the OBD data collected from 3 UPS trucks in Fort Worth Texas between 1/1/2017 and 6/30/2018. There were 10129

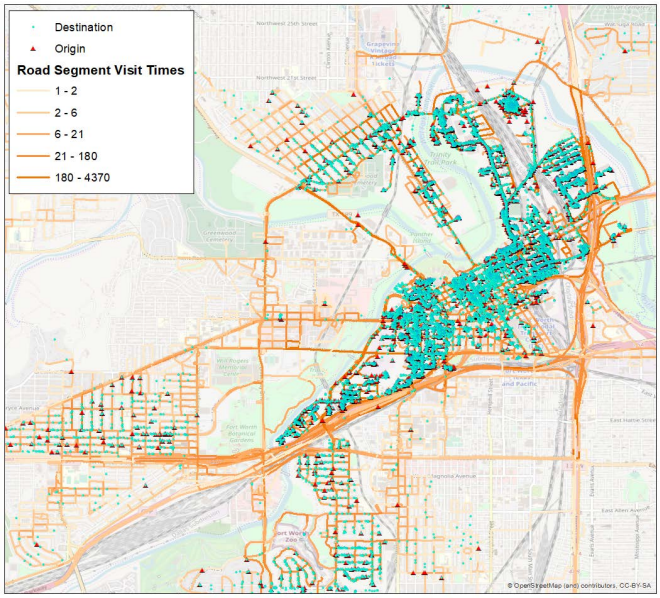


Figure 5.14: A map of the road segments visited by traces in the OBD data and OD pairs.

traces in the OBD data, each of which logged 250 engine measurements (e.g., energy consumption and stop count) along with the geographic location of a vehicle varying with time when it moved between two delivery stops. A map matching algorithm from [122] was used to align the data with a digital map from the OpenStreetMaps, which contained 9084 road segments and 6193 intersections. Figure 5.14 shows the map of Fort Worth, TX with the OBD data. The orange lines show the distribution of OBD data on the map. The darker shades of orange show those paths that had a larger number of traces along them. The origin-destination (OD) pairs of each energy-efficient path query in the experiments were the OD pairs of the traces, so there were 10129 OD pairs in total. In Figure 5.14, origins are the red triangles, while destinations are the blue circles. To calculate the MFPG heuristic in the experiments, the parameters affecting the heuristic were set as $m = 3000kg$, $g = 9.8m/s^2$, $A = 2m^2$, $s = 20miles/hour$, $\rho = 1.14kg/m^3$, $c_{air} = 0.4$ [123], and $c_{rr} = 0.4$ [124].

The spatial graph in the synthetic data consisted of 5929 spatial-nodes, and 9560 spatial-edges between these spatial-nodes in a grid pattern. Spatial-edges were between adjacent spatial nodes (horizontally and vertically) with a probability. The degree of

each spatial-node is 3.2 on average. 15000 traces were randomly created on the graph. The length of the traces was between 1 to 120. For the sake of simplicity, the travel cost metric was time. The speed of the vehicle was assumed to be between 30 and 50 miles per hour. Left turns were given a higher time penalty compared with right turns, to reflect the actual road conditions. Out of the 9560 spatial edges in the graph, 3968 were traversed by at least 1 trace. The OD pairs of each fastest path query in the experiments were the OD pairs of the synthetic traces and 10000 randomly generated OD pairs.

The experiments were conducted on a machine with Intel(R) Core™i5-7500 CPU @ 3.40GHz and 64GB memory. The operating system used was Windows 10. The algorithms were implemented using C# .NET Framework 4.7.

5.5.2 Experiment Results

Energy consumption saving experiments

Among 10129 OD pairs of the traces in the OBD data, the proposed MFPG-LC method found energy-efficient path between 4300 (about 42.45%) of them. Then, we queried the path between these 4300 OD pairs using the GMR Eco-routing method, and estimated the EEC of the result paths using the physics-guided model. Figure 5.15 shows the average EEC of the paths selected by the GMR Eco-routing method and the MFPG-LC method along with the length of the shortest paths between the OD pairs. For example, between the OD pairs between which the shortest paths are of length from 1 km to 2 km, the average EEC of the paths selected by the GMR Eco-routing method is 1.07 kWh, and that of the paths selected by the MFPG-LC method is 0.61 kWh. The average EEC of the paths selected by both methods increased as the OD pairs grew away from each other, but the average EEC of the paths selected the GMR Eco-routing method was always higher than that of the MFPG-LC method. Therefore, the proposed method was able to find paths with lower EEC than the GMR method.

Computational performance experiments

- *Are the proposed algorithms more efficient than the algorithms in preliminary work?*

The number of spatial-edges on all FPs varies with the number of traces (T) in OBD

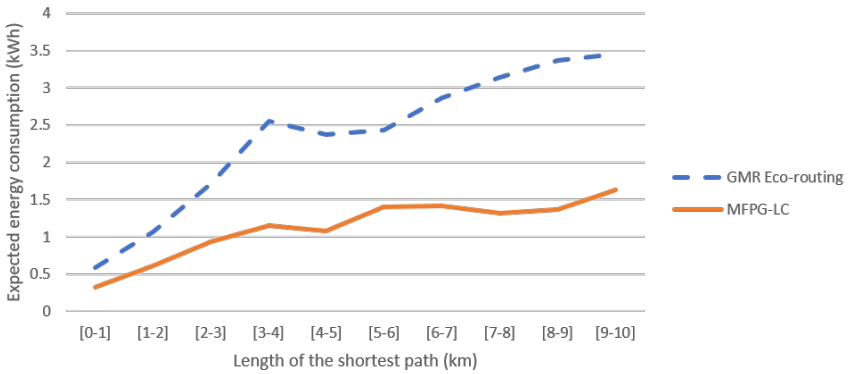


Figure 5.15: The average EEC of the paths selected by the GMR Eco-routing method and the MFPG-LC method.

Table 5.7: Number of frequented spatial-edges varying with the number of traces in OBD data.

(a) Spatial-edges with more than 20 traces along it in the real-world data

Number of Traces	5065	7597	10129
Number of Road Segments	864	961	1112

(b) Spatial-edges with more than 25 traces along it in the synthetic data

Number of Traces	7500	11250	15000
Number of Road Segments	3029	3388	3538

data (Table 5.7), and with the minimum number of traces (β) on an FP (Table 5.8). Hence the number of FPs and UFPs depends on these two factors as well, which affects the search space of the path selection algorithms in turn. The number of edges in the result paths also affects the number of iterations the algorithms need to find the path. Therefore, we compared the performance of the algorithms with a fixed β (20 for the real-world data, and 25 for the synthetic data) and a fixed T (all traces in both datasets) on result paths with varying length.

Figure 5.16 shows the execution time of the algorithms in the case where the travel cost is non-negative. In each sub-figures, the Y axis is the execution time, while the Y axis is the result path length. We can see that in all cases the MFPG-SP and the IN-MFPG-SP algorithms are faster than the physics-guided method. The difference in

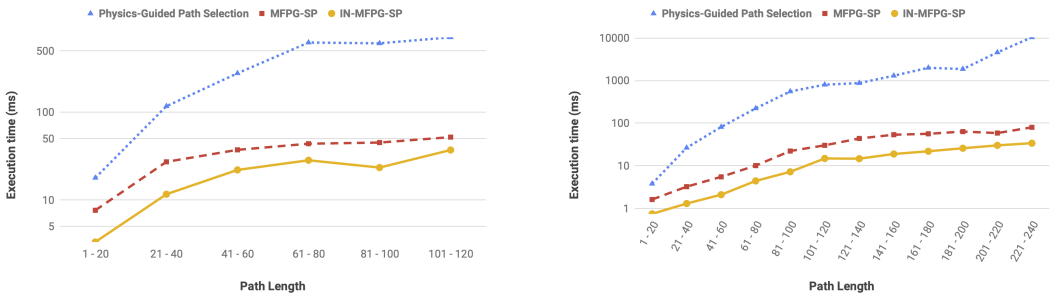
Table 5.8: Number of frequented spatial-edges varying with the minimum number of traces on it.

(a) Real-world data

Minimum number of traces on a FP	20	35	50
Number of Road Segments	1112	858	731

(b) Synthetic data

Minimum number of traces on a FP	25	50	75
Number of Road Segments	3538	2988	2166



(a) Real-world data.

(b) Synthetic data.

Figure 5.16: Is the proposed IN-MFPG-SP algorithm more efficient than the physics-guided, MFPG-SP algorithms?

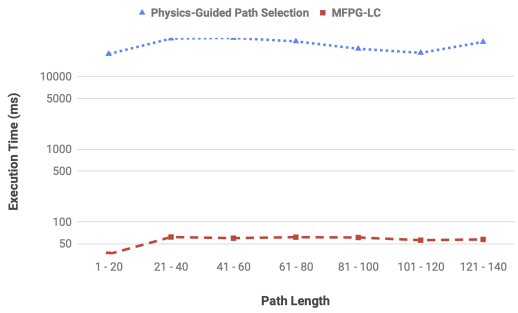


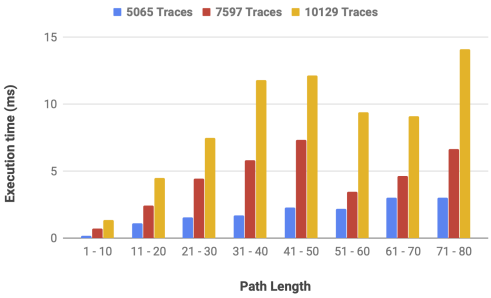
Figure 5.17: Is the proposed MFPG-LC algorithm more efficient than the physics-guided algorithm?

their performance becomes more evident as the path length increases. This is because as the path length increases, the physics-guided method requires more iterations to traverse a given path compared with the MFPG-SP and the IN-MFPG-SP algorithms, which extend paths by appending paths rather than edges. Also, we see that the admissible heuristic reduces the time required by the IN-MFPG-SP algorithm, as the search space is guided towards the destination.

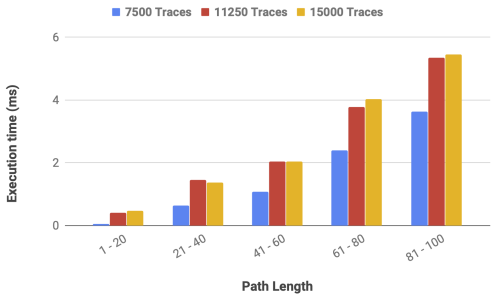
Figure 5.17 shows the execution time of the algorithms in the case where the negative travel cost is allowed. Again, the Y axis is the execution time, while the Y axis is the result path length. We see that the execution times for both the MFPG-LC and the physics-guided algorithms are higher than the case where the travel cost is non-negative in Figure 5.16a. This is because it took these algorithms longer to terminate once a path is found to deal with the negative edge costs. In this case too we find that the MFPG-LC algorithm outperforms the physics-guided algorithm, greatly reducing the amount of time taken by the algorithm.

- *How are the proposed methods affected by the number of input traces?*

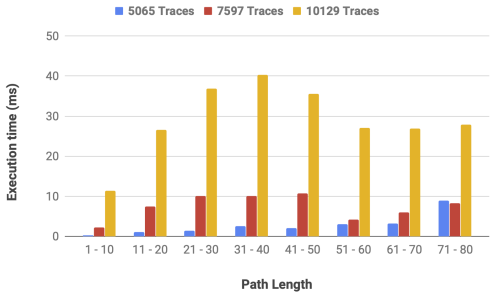
To illustrate the sensitivity of the proposed algorithms on the number of traces in the input data, we generated two subsets of both the real-world and the synthetic data using random sampling without replacement. One had 75% original traces, and the other one had 50% original traces. To avoid the effect of the result path length, the effect of the number of traces in the input data is shown in groups of similar result path length in Figure 5.18, where the average execution time of each path selection query is the y axis,



(a) IN-MFPG-SP algorithm with the real-world data.

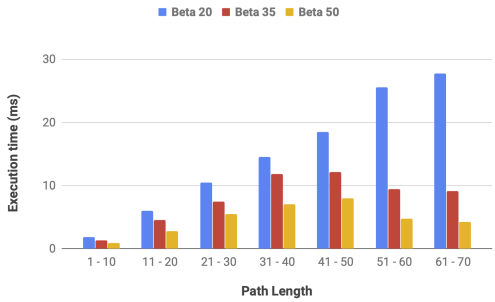


(b) IN-MFPG-SP algorithm with the synthetic data.

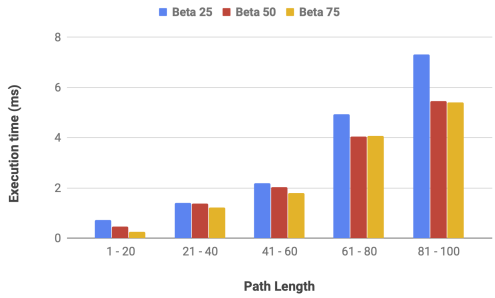


(c) MFPG-LC algorithm with the real-world data.

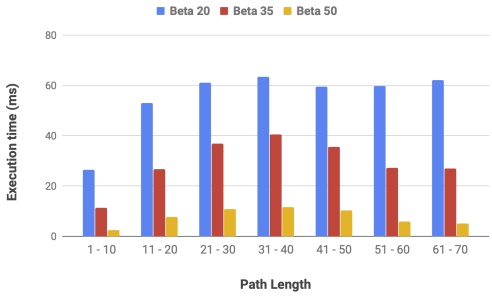
Figure 5.18: How are the proposed methods affected by the number of input traces and the result path length?



(a) IN-MFPG-SP algorithm with the real-world data.



(b) IN-MFPG-SP algorithm with the synthetic data.



(c) MFPG-LC algorithm with the real-world data.

Figure 5.19: How are the proposed methods affected by the minimum number of traces along a FP and the result path length?

and the result path length is the x axis. The execution time on the 50%, 75%, and entire dataset is shown in blue, red, and orange respectively. As we can see, the execution time of all the proposed algorithms increases with the number of input traces. The reason to this phenomenon is that increasing the number of input traces increases the number of FPs and UFPs, and increases the search space of the path selection algorithms in turn.

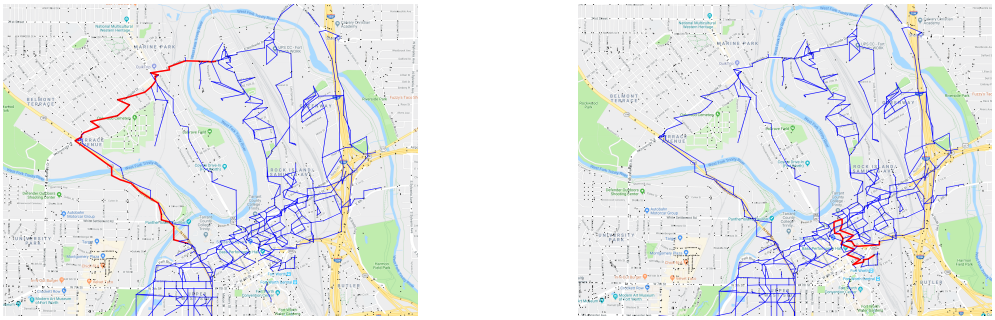
- *How are the proposed methods affected by the minimum number of traces along a FP (β)?*

To show the effect of the minimum number (β) of traces along a frequented path (FP), we executed the proposed algorithms with varying β. $\beta = 20, 35, 50$ for real-world data, and $\beta = 25, 50, 75$ for synthetic data. To avoid the effect of the result path length, the effect of β is shown in groups of similar result path length in Figure 5.19, where the

average execution time of each path selection query is the y axis, and the result path length is the x axis. The execution time with different β is shown in different colors. The results indicate that with β increasing, the execution time of the proposed algorithms stays the same or decreases. As analyzed before, as β increases, the number of road segments that have at least β traces along them decreases, which shrinks the search space of the algorithms in turn. Moreover, the effect of β increases with the length of the result paths. The reason to this is that the value of β affects the number of new candidate paths can be explored at an MFP-node. The difference in the number of new candidate paths with different value of β accumulates with the length of the result path. Hence choosing an appropriate value of β is essential to make these algorithms run efficiently.

- *How are the proposed methods affected by the length of the result path ?*

The length of the result path affects the number of iteration the path selection algorithm needs to go through to find a path. According to both Figure 5.18 and 5.19, in most cases, increasing length of the result path should increase the execution time. Since the input traces are not evenly distributed throughout the spatial graph, some longer paths may have smaller execution time, as seen in Figure 5.18a, between road segment length 50-70, and in Figure 5.18c between road segment lengths 50-80. This is due to some longer paths going along areas that have fewer FPs, resulting in a smaller number of nodes being expanded along these paths. Figure 5.20 shows two such paths from the UPS truck data. The lines in blue indicate the road segments that have at least 35 traces along them, where FPs exist. The red lines indicate the path found for given origin-destination pairs. As can be seen, a lot of the FPs concentrate in a few regions in the road network. Figure 5.20a shows a path composed of 88 edges. This path does not pass through a region where FPs concentrate, hence it expands a relatively fewer number of nodes. To find this path, the IN-MFPG-SP algorithm visited 123 spatial-nodes in total, which cost only 45ms. Figure 5.20b however shows a path that passes through a region where FPs concentrate. Although this path is composed of 16 edges, it took the IN-MFPG-SP algorithm 246ms to find it, and 527 spatial-nodes are visited in the searching process.



(a) Path with length 88 edges

(b) Path with length 16 edges

Figure 5.20: Paths in the real-world data with abnormal computational time

5.6 Case Studies

We conducted two case studies: 1) to estimate the potential energy saving resulting from leveraging the proposed energy-efficient path selection method, and 2) to illustrate that the proposed method can select paths that are more energy-efficient than the paths selected by the currently widely-used path selection methods. The data used in case studies was the same real-world OBD data and road system used in the experiments.

5.6.1 Energy Saving Resulting from the Proposed Method

We queried the energy-efficient paths between 10129 origin and destination (OD) pairs of the traces in the OBD data using the proposed method, and compared their energy consumption with that of the historical paths in the data. Energy-efficient paths are found between 4300 (about 42.45%) OD pairs. The energy-efficient paths between 2510 (about 24.78%) OD pairs have lower expected energy consumption than the historical paths, and the others have the same as the historical paths. Figure 5.21 shows the frequency distribution of the 2510 OD pairs between which the energy-efficient paths have lower expected energy consumption than the historical paths according to the relative difference between the expected energy consumption. For example, the energy-efficient paths between 680 OD pairs save 0-20% energy compared with the historical paths between the OD pairs. In this case study, the energy saving resulted from leveraging the proposed method was about 12.10% of the total energy consumption of the traces in the OBD data.

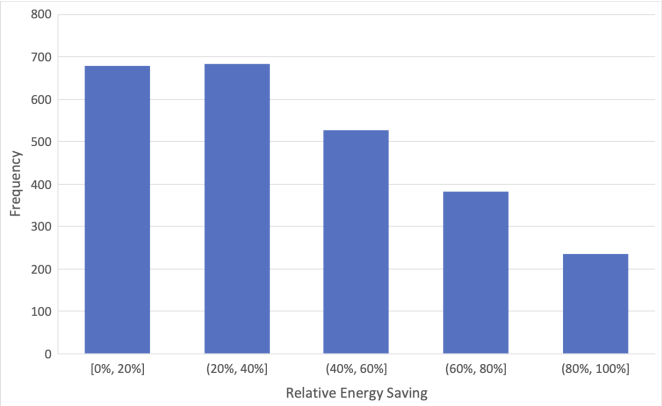


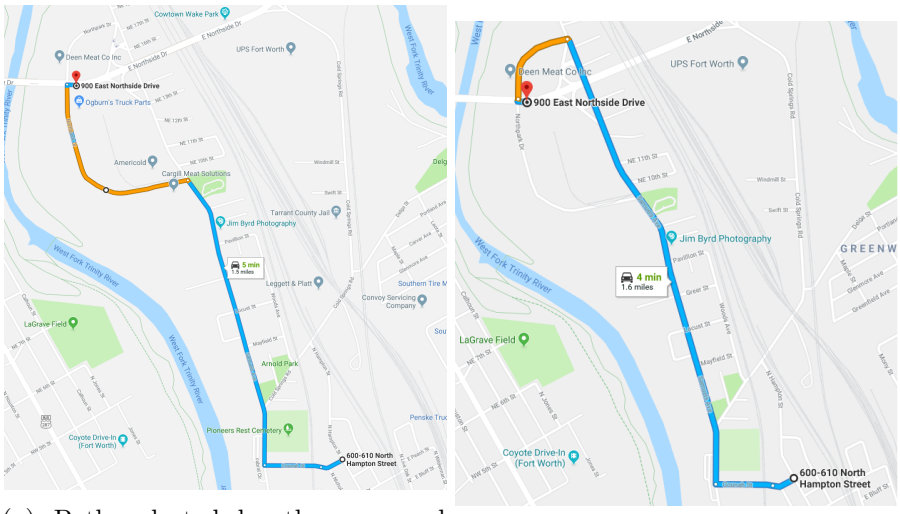
Figure 5.21: Frequency distribution of the OD pairs according to the relative difference between the expected energy consumption on the energy-efficient paths and that on the historical paths between them.

5.6.2 Comparison between the Proposed Method and Google Maps

We searched for a path between two road intersections in Fort Worth, TX using the proposed method and Google Maps, which was an example of the popular tools for routing. The proposed method selected a path 1.5 miles long with an estimated time cost of 5 minutes (Figure 5.22a). Google Maps chose a path 1.6 miles long but with a smaller estimated time cost of 4 minutes (Figure 5.22b). Nevertheless, the path selected by the proposed method had a lower estimated energy cost (1.11 KWh) than the path from Google Maps (1.60 KWh). A potential cause of this difference was that the part of the route affected by heavy traffic (shown in orange) was longer on the path selected by the proposed method than on the Google Maps path, but its impact on energy consumption was not as large as on time cost. Therefore, the proposed method was able to select the more energy-efficient path than the currently widely-used method.

5.7 A Road Test in Cincinnati, OH

We also conducted a real-world road test using one UPS delivery truck and one driver. Figure 5.23 shows the overview of the study area. The task was to find a path between 100 Commerce Dr, Loveland, OH and 8063 Montgomery Road, Cincinnati, OH. Google Maps suggested the path highlighted in blue. Our proposed method, instead, suggested



(a) Path selected by the proposed method.

(b) Path selected by Google Maps.

Figure 5.22: A path selected by the proposed method is more energy-efficient than that from Google Maps.

the path highlighted in green, and estimated that the expected energy consumption of the blue path found by Google Maps was 8.54 kWh, and that of the green path suggested by the proposed method was 5.43 kWh. The expected time costs of the blue and green paths are 14 and 17 minutes respectively. An explanation of this result is that the main segment of the blue path is an U.S. interstate, while that of the green path is an U.S. state highway, so the blue path has a greater speed limit than the green path. In addition, the length of the blue and green paths is 9.2 and 7.6 miles respectively. Even though the high speed on the interstate reduces the travel time, it causes high energy consumption. Then, the test truck was driven on both paths twelve times to collect data for validation. According to the validation data, the average energy consumption on the two paths are 8.27 kWh, and 5.09 kWh. Therefore, the proposed method was able to select the more energy-efficient path than the currently widely-used method.

5.8 Conclusion and Future Work

Today's increasing volume of OBD data facilitates monitoring and managing traffic and transportation systems using the data from connected vehicles, which is an important

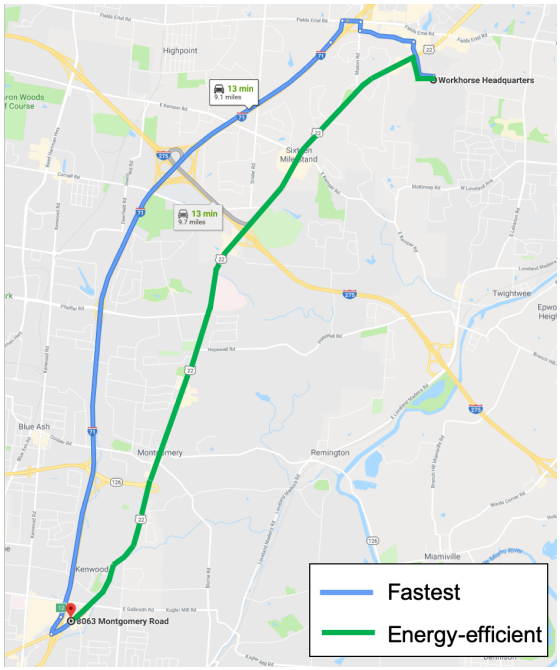


Figure 5.23: Paths suggested by Google Maps and the proposed method.

component of a smart city. In this chapter, we explored the energy-efficient path selection problem, whose challenges included the dependence of the energy consumption on the physical parameters of vehicles, the dependence of the energy consumption on different edges along a path, the high computational cost of estimating expected energy consumption on a path, and the potentially negative energy consumption. We proposed the MFPG heuristic that can guide the search space of the path selection algorithms towards the destination and still guarantee finding the correct path. We also introduced an IN-MFPG-SP algorithm that uses the MFPG heuristic. In addition, we proposed the MFPG-LC algorithm to handle possibly negative energy consumption using a label correcting strategy. We analyzed the proposed algorithms for correctness, completeness and computational time complexity. The experiments we conducted on both real-world data and synthetic data showed that the proposed algorithms yielded substantial computational savings compared to the algorithms in our preliminary work. Then, we conducted two case studies, which illustrated that leveraging the proposed method would save 12.10% energy consumption, and that the path selected by the

proposed method was more energy-efficient than the path found by Google Maps and the state-of-the-art eco-routing method. Last, we conducted a real-world road test to validate that the proposed method can help to save energy compared with Google Maps.

There are several avenues we can pursue in future work, including the precomputation and the storage of the maximal frequented path graph (MFPG), updating the MFPG given changing OBD data rather than regenerating the MFPG every time the OBD data are updated.

Chapter 6

Conclusions and Future Work

6.1 Key Results

The increasing amount of available spatial data over the last decade boosts the need for geospatial artificial intelligent (GeoAI) techniques to solve the challenges posed by the data, including the gap between conventional AI techniques and domain knowledge and challenges caused by the properties of spatial data. This thesis addressed some of these challenges for four groups of GeoAI applications (i.e., descriptive, diagnostic, predictive, prescriptive) using spatial data in two common data types (i.e., point sets and multi-attributed trajectories). First, the thesis proposed a local colocation pattern detection method to detect spatial colocation patterns that may not be prevalent globally but prevalent in regions because of spatial heterogeneity. The thesis introduces a Quad & Grid filter-refine algorithm to accelerate the computation without affecting the correctness and completeness of the results. Second, the thesis investigate the problem of discovering contrasting spatial colocation patterns that have different prevalence in two groups of spatial datasets. It leverages the domain knowledge that neighborhood relationships between categorical spatial objects may convey important information, and introduces a filter & refine algorithm using the anti-monotone property of a proposed metric to measure the prevalence difference of any colocation patterns in the two groups. Third, the thesis discusses a point-set classification method for multiplexed pathology images. Inspired by the domain assumption that the spatial configuration of cells may vary under different health conditions, this thesis introduces a neural network architecture to capture the

Table 6.1: Thesis contribution taxonomy.

		GeoAI tasks			
		Descriptive	Diagnostic	Predictive	Prescriptive
Data type	Point set	Local colocation (chapter 2), Point set representation learning	Contrasting colocation (chapter 3), Contrasting colocation in multiple classes of point sets	Point set classification (chapter 4)	
	Multi-attributed trajectory			Physics-informed energy consumption estimation, Energy consumption probability distribution estimation	Eco-routing (chapter 5)
	...				

spatial configurations of categorical point sets through modeling pairwise relationships. Last, the thesis introduces a physics-guided K-means algorithms to estimate the energy consumption for a vehicle to travel along a path, which is a combination of physics laws followed by vehicle energy consumption and a machine learning model. The thesis also proposes a path-centric path selection algorithm using the proposed energy consumption estimation model considering the spatial autocorrelation property of the data.

Table 6.1 summarizes these key results, and also outlines both short-term and long-term future work.

6.2 Short Term Future Directions

In the short-term, as shown in Table 6.1, we plan to investigate (1) expected energy consumption estimation using onboard diagnostics data from vehicles and (2) contrasting spatial colocation pattern detection in more than two groups of spatial datasets.

- Given a road network and historical on-board diagnostics data from vehicles with known physical parameters, the physics-guided K-means method introduced in this thesis can only estimate the expected traveling energy consumption on paths with enough historical data, because as an integration of a physical law and an unsupervised machine learning model, the proposed method estimated the energy consumption on a path only using the data on the path. Therefore, we plan to develop a physics-guided deep neural network method to significantly increase the applicable scenarios of the method. First of all, deep neural networks such as convolutional neural network (CNN) and recurrent neural network (RNN) have been widely used to capture spatial dependency. In addition, even though onboard diagnostics data from vehicles are growing available with the popularity of telematics devices equipped with GPS chips, the amount of data with energy consumption information is still much less than those with traveling time information. By leveraging data with traveling time information and deep neural network, we plan to explore a model that can be applied on paths with or without historical data with energy consumption information.

- We plan to explore contrasting spatial colocation pattern detection in multiple groups of spatial datasets, which generalize the problem of detecting contrasting spatial colocation pattern detection in two groups of datasets that is discussed in this thesis. The generalized problem is of significant societal importance because there are lots of application cases where the contrasting spatial colocation patterns in multiple classes of datasets are of interest. For example, pathologists may be interested in the neighborhood relationships of cells in multiple disease stages. Biologists may be interested in the different colocation patterns in multiple climate zones. The challenges of the problem are two-fold. First, To describe the difference between the prevalence of the spatial colocation patterns in multiple groups of datasets, we will need to propose a novel statistic.

6.3 Long Term Future Directions

In the long term, as shown in Table 6.1, we will attempt to investigate (1) energy consumption probability distribution estimation using onboard diagnostics data from vehicles, (2) representation learning of spatial relationships in categorical point set.

- Given a road network and historical on-board diagnostics (OBD) data from vehicles with known physical parameters, the energy consumption probability distribution estimation problem aims to project the probability distribution of the energy consumption of a vehicle to travel along a path. The importance of this problem is because the variance of vehicles' energy consumption on a path may be large, which makes the expected energy consumption less representative. By estimating the probability distribution of the energy consumption, we will be able to predict the chance that a vehicle can reach a destination with certain amount of energy, which is can relief drivers' range anxiety especially for electric car drivers. The challenges of this problem include the limited availability of OBD data, the dependency of energy consumption on vehicles' physical parameters, and modeling the probability distribution. We plan to design a physics-informed neural network framework to predict the distribution.

- In our current work, we mainly use the spatial colocation pattern to model the proximity relationships in a multi-categorical point set. However, there are many other more complicated spatial relationships in point sets such as surrounding, within, etc, which may be useful for downstream tasks involving multi-categorical point sets such as classification, prediction. The representation learning of spatial relationships problem aims to train a model that can capture these relationships. The first challenge of this problem is that unlike tabular records, images, and text that are commonly used as the input data in machine learning problems, point sets are in irregular form. In addition, there may be a large number of types of spatial relationships, which are in complex forms, which may require a large volume of data to train a model to learn the representation of the relationships. We plan to integrate the domain knowledge into machine learning models to mitigate the need for data and simplify the machine learning models that are needed to solve the problems.

References

[1] Yiqun Xie, Emre Eftelioglu, Reem Y Ali, Xun Tang, Yan Li, Ruhi Doshi, and Shashi Shekhar. Transdisciplinary foundations of geospatial data science. *ISPRS International Journal of Geo-Information*, 6(12):395, 2017.

[2] Hong-Sen Yan and marco ceccarelli, editors. *International Symposium on History of Machines and Mechanisms: Proceedings of HMM 2008*. History of Mechanism and Machine Science. Springer Netherlands, 2009.

[3] James B Campbell and Randolph H Wynne. *Introduction to remote sensing*. Guilford Press, 2011.

[4] Neha Joshi, Matthias Baumann, Andrea Ehammer, Rasmus Fensholt, Kenneth Grogan, Patrick Hostert, Martin Rudbeck Jepsen, Tobias Kuemmerle, Patrick Meyfroidt, Edward T. A. Mitchard, Johannes Reiche, Casey M. Ryan, and Björn Waske. A Review of the Application of Optical and Radar Remote Sensing Data Fusion to Land Use Mapping and Monitoring. *Remote Sensing*, 8(1):70, January 2016.

[5] S. Sinha, C. Jeganathan, L. K. Sharma, and M. S. Nathawat. A review of radar remote sensing for biomass estimation. *International Journal of Environmental Science and Technology*, 12(5):1779–1792, May 2015.

[6] Bin Wu, Bailang Yu, Qiusheng Wu, Shenjun Yao, Feng Zhao, Weiqing Mao, and Jianping Wu. A Graph-Based Approach for 3d Building Model Reconstruction from Airborne LiDAR Point Clouds. *Remote Sensing*, 9(1):92, January 2017.

[7] Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. Semantic Trajectories: Mobility Data Computation and Annotation. *ACM Trans. Intell. Syst. Technol.*, 4(3):49:1–49:38, July 2013.

[8] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You Are Where You Tweet: A Content-based Approach to Geo-locating Twitter Users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 759–768, New York, NY, USA, 2010. ACM. event-place: Toronto, ON, Canada.

[9] Yan Li and Shashi Shekhar. Local co-location pattern detection: a summary of results. In *10th International Conference on Geographic Information Science (GIScience 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[10] Emre Eftelioglu, Shashi Shekhar, James M Kang, and Christopher C Farah. Ring-shaped hotspot detection. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3367–3381, 2016.

[11] Yan Li, Pratik Kotwal, Pengyue Wang, Shashi Shekhar, and William Northrop. Trajectory-aware lowest-cost path selection: A summary of results. In *Proceedings of the 16th International Symposium on Spatial and Temporal Databases*, pages 61–69, 2019.

[12] Yan Li, Shashi Shekhar, Pengyue Wang, and William Northrop. Physics-guided energy-efficient path selection: a summary of results. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 99–108, 2018.

[13] Yuhao Kang, Song Gao, Yunlei Liang, Mingxiao Li, Jinmeng Rao, and Jake Kruse. Multiscale dynamic human mobility flow dataset in the us during the covid-19 epidemic. *Scientific data*, 7(1):1–13, 2020.

[14] Krzysztof Janowicz, Song Gao, Grant McKenzie, Yingjie Hu, and Budhendra Bhaduri. Geoai: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond, 2020.

[15] Yiqun Xie, Han Bao, Shashi Shekhar, and Joseph Knight. A timber framework for mining urban tree inventories using remote sensing datasets. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1344–1349. IEEE, 2018.

[16] Wenwen Li and Chia-Yu Hsu. Automated terrain feature identification from remote sensing imagery: a deep learning approach. *International Journal of Geographical Information Science*, 34(4):637–660, 2020.

[17] Emre Eftelioglu, Yan Li, Xun Tang, Shashi Shekhar, James M Kang, and Christopher Farah. Mining network hotspots with holes: A summary of results. In *The Annual International Conference on Geographic Information Science*, pages 51–67. Springer, 2016.

[18] Yiqun Xie, Shashi Shekhar, and Yan Li. Statistically-robust clustering techniques for mapping spatial hotspots: A survey. *arXiv preprint arXiv:2103.12019*, 2021.

[19] Yan Li, Yiqun Xie, Pengyue Wang, Shashi Shekhar, and William Northrop. Significant lagrangian linear hotspot discovery. In *Proceedings of the 13th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pages 1–10, 2020.

[20] Dong Wang, Nima Nejadsadeghi, Yan Li, Shashi Shekhar, Anil Misra, and Joshua A Dijksman. Rotational diffusion and rotational correlations in frictional amorphous disk packings under shear. *Soft matter*, 17(34):7844–7852, 2021.

[21] Yan Li, Pratik Kotwal, Pengyue Wang, Yiqun Xie, Shashi Shekhar, and William Northrop. Physics-guided energy-efficient path selection using on-board diagnostics data. *ACM Transactions on Data Science*, 1(3):1–28, 2020.

[22] DIANE COOK and LEN JENSHEL. Buying Guide - Cars and Their Environmental Impact, January 2017.

[23] U.S. Energy Information Administration. Total u.s. energy expenditures in 2015 were the lowest in more than a decade, August 2017.

[24] Biopsy, Aug 2020.

[25] Yury Goltsev, Nikolay Samusik, Julia Kennedy-Darling, Salil Bhate, Matthew Hale, Gustavo Vazquez, Sarah Black, and Garry P. Nolan. Deep profiling of mouse splenic architecture with codex multiplexed imaging. 2017.

[26] Jia-Ren Lin, Benjamin Izar, Shu Wang, Clarence Yapp, Shaolin Mei, Parin M Shah, Sandro Santagata, and Peter K Sorger. Highly multiplexed immunofluorescence imaging of human tissues and tumors using t-cycif and conventional optical microscopes. *eLife*, 7, 2018.

[27] Steve Lu, Julie E. Stein, David L. Rimm, Daphne W. Wang, J. Michael Bell, Douglas B. Johnson, Jeffrey A. Sosman, Kurt A. Schalper, Robert A. Anders, Hao Wang, and et al. Comparison of biomarker modalities for predicting response to pd-1/pd-l1 checkpoint blockade. *JAMA Oncology*, 5(8):1195, 2019.

[28] Souptik Barua, Penny Fang, Amrish Sharma, Junya Fujimoto, Ignacio Wistuba, Arvind U.K. Rao, and Steven H. Lin. Spatial interaction of tumor cells and regulatory t cells correlates with survival in non-small cell lung cancer. *Lung Cancer*, 117:73–79, 2018.

[29] Iwona Lugowska, Pawel Teterycz, and Piotr Rutkowski. Immunotherapy of melanoma. *Współczesna Onkologia*, 2018(1):61–67, 2018.

[30] K Noel Masihi. Fighting infection using immunomodulatory agents. *Expert Opinion on Biological Therapy*, 1(4):641–653, 2001.

[31] Hongming Zhang and Jibei Chen. Current status and future directions of cancer immunotherapy. *Journal of Cancer*, 9(10):1773–1781, 2018.

[32] Yinyin Yuan. Spatial heterogeneity in the tumor microenvironment. *Cold Spring Harbor Perspectives in Medicine*, 6(8), 2016.

[33] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 1(1):1–34, 2020.

[34] W. R. Tobler. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 46(sup1):234–240, June 1970.

[35] Shashi Shekhar, Zhe Jiang, Reem Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. Spatiotemporal data mining: A computational perspective. *ISPRS International Journal of Geo-Information*, 4(4):2306–2338, 2015.

[36] Z. Jiang, S. Shekhar, X. Zhou, J. Knight, and J. Corcoran. Focal-Test-Based Spatial Decision Tree Learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1547–1559, June 2015.

[37] Jayant Gupta, Carl Molnar, Yiqun Xie, Joe Knight, and Shashi Shekhar. Spatial variability aware deep neural networks (svann): A general approach. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(6):1–21, 2021.

[38] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9596–9606, 2019.

[39] David Daley. GOP Racial Gerrymandering Mastermind Participated in Redistricting in More States Than Previously Known, Files Reveal, September 2019.

[40] Alexander J. Stewart, Mohsen Mosleh, Marina Diakonova, Antonio A. Arechar, David G. Rand, and Joshua B. Plotkin. Information gerrymandering and undemocratic decisions. *Nature*, 573(7772):117–121, September 2019.

[41] Yan Li, Majid Farhadloo, Santhoshi Krishnan, Timothy L Frankel, Shashi Shekhar, and Arvind Rao. Srnet: A spatial-relationship aware point-set classification method for multiplexed pathology images. In *Proceedings of DeepSpatial'21: 2nd ACM SIGKDD Workshop on Deep Learning for Spatiotemporal Data, Applications, and Systems*, volume 10, 2021.

[42] Guilbert Gates, Jack Ewing, Karl Russell, and Derek Watkins. How Volkswagen's 'Defeat Devices' Worked. *The New York Times*, October 2015.

[43] Pradeep Mohan, Shashi Shekhar, James A. Shine, James P. Rogers, Zhe Jiang, and Nicole Wayant. A Neighborhood Graph Based Approach to Regional Co-location Pattern Discovery: A Summary of Results. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 122–132, New York, NY, USA, 2011. ACM.

[44] Jordan Wood. Minimum Bounding Rectangle. In Shashi Shekhar, Hui Xiong, and Xun Zhou, editors, *Encyclopedia of GIS*, pages 1232–1233. Springer International Publishing, 2 edition, 2017. DOI: 10.1007/978-3-319-17885-1_783.

[45] Yan Huang, Shashi Shekhar, and Hui Xiong. Discovering colocation patterns from spatial data sets: a general approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1472–1485, 2004.

[46] Mete Celik, James M. Kang, and Shashi Shekhar. Zonal co-location pattern discovery with dynamic parameters. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 433–438. IEEE, 2007.

[47] Song Wang, Yan Huang, and Xiaoyang Sean Wang. Regional Co-locations of Arbitrary Shapes. In *Advances in Spatial and Temporal Databases*, pages 19–37. Springer Berlin Heidelberg, August 2013. DOI: 10.1007/978-3-642-40235-7_2.

[48] Christoph F. Eick, Rachana Parmar, Wei Ding, Tomasz F. Stepinski, and Jean-Philippe Nicot. Finding Regional Co-location Patterns for Sets of Continuous Variables in Spatial Datasets. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '08, pages 30:1–30:10, New York, NY, USA, 2008. ACM.

[49] Min Deng, Jiannan Cai, Qiliang Liu, Zhanjun He, and Jianbo Tang. Multi-level method for discovery of regional co-location patterns. *International Journal of Geographical Information Science*, 31(9):1846–1870, September 2017.

[50] Jin Soung Yoo and S. Shekhar. A Joinless Approach for Mining Spatial Colocation Patterns. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1323–1337, October 2006.

[51] Chicago Police Department. Crimes - 2001 to present, 2017. [Online; accessed 30-September-2017].

[52] USGS. North america rivers and lakes, 2018. [Online; accessed 13-February-2018].

[53] Mining Statistically Significant Co-location and Segregation Patterns. 26.

[54] Adrian Baddeley. Spatial Point Processes and their Applications. In *Stochastic Geometry*, volume 1892 of *Lecture Notes in Mathematics*, pages 1–75. Springer, Berlin, Heidelberg, 2007. DOI: 10.1007/978-3-540-38175-4_1.

[55] Sung Nok Chiu, Dietrich Stoyan, Wilfrid S. Kendall, and Joseph Mecke. *Stochastic Geometry and Its Applications*. John Wiley & Sons, June 2013.

[56] Christoph F Eick, Rachana Parmar, Wei Ding, Tomasz F Stepinski, and Jean-Philippe Nicot. Finding regional co-location patterns for sets of continuous variables in spatial datasets. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–10, 2008.

[57] Jiannan Cai, Qiliang Liu, Min Deng, Jianbo Tang, and Zhanjun He. Adaptive detection of statistically significant regional spatial co-location patterns. *Computers, Environment and Urban Systems*, 68:53–63, 2018.

[58] Yan Huang, Shashi Shekhar, and Hui Xiong. Discovering colocation patterns from spatial data sets: a general approach. *IEEE Transactions on Knowledge and data engineering*, 16(12):1472–1485, 2004.

[59] Jin Soung Yoo and Shashi Shekhar. A joinless approach for mining spatial colocation patterns. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1323–1337, 2006.

[60] Lizhen Wang, Yuzhen Bao, and Zhongyu Lu. Efficient discovery of spatial co-location patterns using the icpi-tree. *The Open Information Systems Journal*, 3(1), 2009.

[61] Xiaojing Yao, Ling Peng, Liang Yang, and Tianhe Chi. A fast space-saving algorithm for maximal co-location pattern mining. *Expert Systems with Applications*, 63:310–323, 2016.

[62] Xuguang Bao and Lizhen Wang. A clique-based approach for co-location pattern mining. *Information Sciences*, 490:244–264, 2019.

[63] Arpan Man Sainju and Zhe Jiang. Grid-based colocation mining algorithms on gpu for big spatial event data: A summary of results. In *International Symposium on Spatial and Temporal Databases*, pages 263–280. Springer, 2017.

[64] Peizhong Yang, Lizhen Wang, and Xiaoxuan Wang. A mapreduce approach for spatial co-location pattern mining via ordered-clique-growth. *Distributed and Parallel Databases*, 38(2):531–560, 2020.

[65] Jin Soung Yoo, Douglas Boulware, and David Kimmey. Parallel co-location mining with mapreduce and nosql systems. *Knowledge and Information Systems*, 62(4):1433–1463, 2020.

[66] Yong Ge, Zijun Yao, and Huayu Li. Computing co-location patterns in spatial data with extended objects: a scalable buffer-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[67] Zhiping Ouyang, Lizhen Wang, and Pingping Wu. Spatial co-location pattern discovery from fuzzy objects. *International Journal on Artificial Intelligence Tools*, 26(02):1750003, 2017.

[68] Xiaoxuan Wang, Le Lei, Lizhen Wang, Peizhong Yang, and Hongmei Chen. Spatial co-location pattern discovery incorporating fuzzy theory. *IEEE Transactions on Fuzzy Systems*, 2021.

[69] Sajib Barua and Jörg Sander. Mining statistically significant co-location and segregation patterns. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1185–1199, 2013.

[70] Jiannan Cai, Yiqun Xie, Min Deng, Xun Tang, Yan Li, and Shashi Shekhar. Significant spatial co-distribution pattern discovery. *Computers, Environment and Urban Systems*, 84:101543, 2020.

[71] Zhiyuan Li, Dan Li, Andy Tsun, and Bin Li. Foxp3 regulatory t cells and their functional regulation. *Cellular & Molecular Immunology*, 12(5):558–565, 2015.

[72] Bagher Farhood, Masoud Najafi, and Keywan Mortezaee. Cd8 cytotoxic t lymphocytes in cancer immunotherapy: A review. *Journal of Cellular Physiology*, 234(6):8509–8521, 2018.

[73] Anita Feichtenbeiner, Matthias Haas, Maike Büttner, Gerhard G. Grabenbauer, Rainer Fietkau, and Luitpold V. Distel. Critical role of spatial interaction between cd8 and foxp3 cells in human gastric cancer: the distance matters. *Cancer Immunology, Immunotherapy*, 63(2):111–119, 2013.

[74] Magdalena Huber, Corinna U. Brehm, Thomas M. Gress, Malte Buchholz, Bilal Alashkar Alhamwe, Elke Von Strandmann, Emily P. Slater, Jörg W. Bartsch, Christian Bauer, Matthias Lauth, and et al. The immune microenvironment in pancreatic cancer. *International Journal of Molecular Sciences*, 21(19):7307, 2020.

[75] Wei Chang Colin Tan, Sanjna Nilesh Nerurkar, Hai Yun Cai, Harry Ho Man Ng, Duoduo Wu, Yu Ting Felicia Wee, Jeffrey Chun Tatt Lim, Joe Yeong, and Tony Kiat Hon Lim. Overview of multiplex immunohistochemistry/immunofluorescence techniques in the era of cancer immunotherapy. *Cancer Communications*, 40(4):135–153, 2020.

[76] Babak Ehteshami Bejnordi, Maeve Mullooly, Ruth M Pfeiffer, Shaoqi Fan, Pamela M Vacek, Donald L Weaver, Sally Herschorn, Louise A Brinton, Bram van Ginneken, Nico Karssemeijer, et al. Using deep convolutional neural networks to identify and classify tumor-associated stroma in diagnostic breast biopsies. *Modern Pathology*, 31(10):1502–1512, 2018.

[77] Nicolas Coudray, Paolo Santiago Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre L Moreira, Narges Razavian, and Aristotelis Tsirigos. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nature medicine*, 24(10):1559–1567, 2018.

[78] Eirini Arvaniti, Kim S Fricker, Michael Moret, Niels Rupp, Thomas Hermanns, Christian Fankhauser, Norbert Wey, Peter J Wild, Jan H Rueschoff, and Manfred Claassen. Automated gleason grading of prostate cancer tissue microarrays via deep learning. *Scientific reports*, 8(1):1–11, 2018.

[79] Hrushikesh Garud, Sri Phani Krishna Karri, Debdoot Sheet, Jyotirmoy Chatterjee, Manjunatha Mahadevappa, Ajoy K Ray, Arindam Ghosh, and Ashok K Maity. High-magnification multi-views based classification of breast fine needle aspiration cytology cell samples using fusion of decisions from deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 76–81, 2017.

[80] Ling Zhang, Le Lu, Isabella Nogues, Ronald M Summers, Shaoxiong Liu, and Jianhua Yao. Deeppap: deep convolutional networks for cervical cell classification. *IEEE journal of biomedical and health informatics*, 21(6):1633–1643, 2017.

[81] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[82] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5099–5108, 2017.

[83] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017.

[84] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pages 105–122. Springer, 2018.

[85] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. RGCNN: regularized graph CNN for point cloud segmentation. In Susanne Boll, Kyoung Mu Lee, Jiebo Luo,

Wenwu Zhu, Hyeran Byun, Chang Wen Chen, Rainer Lienhart, and Tao Mei, editors, *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, pages 746–754. ACM, 2018.

[86] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 984–993. IEEE Computer Society, 2018.

[87] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[88] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 8159–8170. IEEE, 2019.

[89] Timothy F Leslie, Cara L Frankenfeld, and Matthew A Makara. The spatial food environment of the dc metropolitan area: Clustering, co-location, and categorical differentiation. *Applied Geography*, 35(1-2):300–307, 2012.

[90] Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-scale representation learning for spatial feature distributions using grid cells. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[91] Ruiqi Gao, Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Learning grid cells as vector representation of self-position coupled with matrix representation of self-motion. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[92] Alison Abbott and Ewen Callaway. Nobel prize for decoding brain's sense of place. *Nature News*, 514(7521):153, 2014.

[93] Sajib Barua and Jörg Sander. Mining statistically significant co-location and segregation patterns. *IEEE Trans. Knowl. Data Eng.*, 26(5):1185–1199, 2014.

[94] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[95] Department of Energy. Energy Department Announces $58 Million to Advance Fuel-Efficient Vehicle Technologies, January 2016.

[96] U.S. Energy Information Administration. International Energy Outlook 2017. Technical Report DOE/EIA-0484(2017), Washington, DC, September 2017.

[97] Kyoungho Ahn and Hesham A. Rakha. Network-wide impacts of eco-routing strategies: A large-scale case study. *Transportation Research Part D: Transport and Environment*, 25(0), December 2013.

[98] Xianan Huang and Huei Peng. Eco-Routing based on a Data Driven Fuel Consumption Model. *arXiv:1801.08602 [stat]*, January 2018. arXiv: 1801.08602.

[99] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity, May 2011.

[100] A. Cappiello, I. Chabini, E. K. Nam, A. Lue, and M. Abou Zeid. A statistical model of vehicle emissions and fuel consumption. In *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 801–809, 2002.

[101] J. Kwon, A. Rousseau, and P. Sharer. Analyzing the Uncertainty in the Fuel Economy Prediction for the EPA MOVES Binning Methodology. SAE Technical Paper 2007-01-0280, SAE International, Warrendale, PA, April 2007.

[102] Ram Vijayagopal, Larry Michaels, Aymeric P. Rousseau, Shane Halbach, and Neeraj Shidore. Automated Model Based Design Process to Evaluate Advanced

Component Technologies. SAE Technical Paper 2010-01-0936, SAE International, Warrendale, PA, April 2010.

[103] Aaron Brooker, Jeffrey Gonder, Lijuan Wang, Eric Wood, Sean Lopp, and Laurie Ramroth. FASTSim: A Model to Estimate Vehicle Efficiency, Cost and Performance. SAE Technical Paper 2015-01-0973, SAE International, Warrendale, PA, April 2015.

[104] L. Zhu, J. Holden, E. Wood, and J. Gonder. Green routing fuel saving opportunity assessment: A case study using large-scale real-world travel data. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1242–1248, June 2017.

[105] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.

[106] P. E. Hart, N. J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[107] USDOE. ARPA-E | NEXTCAR, November 2016.

[108] EU Horizon 2020 Research and Innovation Programme. optiTruck.

[109] Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer, December 2008.

[110] Christian Sommer. Shortest-path Queries in Static Networks. *ACM Comput. Surv.*, 46(4):45:1–45:31, March 2014.

[111] Sabeur Aridhi, Philippe Lacomme, Libo Ren, and Benjamin Vincent. A MapReduce-based approach for shortest path problem in large-scale networks. *Engineering Applications of Artificial Intelligence*, 41:151–165, May 2015.

[112] Daniel Delling, Andrew V. Goldberg, Andreas Nowatzyk, and Renato F. Werneck. PHAST: Hardware-accelerated shortest path trees. *Journal of Parallel and Distributed Computing*, 73(7):940–952, July 2013.

[113] Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. The Shortest Path Problem Revisited: Optimal Routing for Electric Vehicles. In *KI 2010: Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 309–316. Springer, Berlin, Heidelberg, September 2010.

[114] Jochen Eisner, Stefan Funke, and Sabine Storandt. Optimal Route Planning for Electric Vehicles in Large Networks. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 1108–1113, San Francisco, California, 2011. AAAI Press.

[115] Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy Routes in the City. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, HT '14, pages 116–125, New York, NY, USA, 2014. ACM.

[116] Daniel Duque, Leonardo Lozano, and Andrés L. Medaglia. An exact method for the biobjective shortest path problem for large-scale road networks. *European Journal of Operational Research*, 242(3):788–797, May 2015.

[117] V. M. V. Gunturi, S. Shekhar, and K. Yang. A Critical-Time-Point Approach to All-Departure-Time Lagrangian Shortest Paths. *IEEE Transactions on Knowledge and Data Engineering*, 27(10):2591–2603, October 2015.

[118] B. Y. Chen, W. H. K. Lam, Q. Li, A. Sumalee, and K. Yan. Shortest Path Finding Problem in Stochastic Time-Dependent Road Networks With Stochastic First-In-First-Out Property. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1907–1917, December 2013.

[119] Yuan Gao. Shortest path problem with uncertain arc lengths. *Computers & Mathematics with Applications*, 62(6):2591–2600, September 2011.

[120] Yong Deng, Yuxin Chen, Yajuan Zhang, and Sankaran Mahadevan. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*, 12(3):1231–1237, March 2012.

[121] Bin Yang, Jian Dai, Chenjuan Guo, Christian S. Jensen, and Jilin Hu. PACE: a PAth-CEntric paradigm for stochastic path finding. *The VLDB Journal*, 27(2):153–178, April 2018.

[122] Paul Newson and John Krumm. Hidden Markov Map Matching Through Noise and Sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 336–343, New York, NY, USA, 2009. ACM. event-place: Seattle, Washington.

[123] Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. The shortest path problem revisited: Optimal routing for electric vehicles. In *Annual conference on artificial intelligence*, pages 309–316. Springer, 2010.

[124] Heinz Heisler. 14 - vehicle body aerodynamics. In Heinz Heisler, editor, *Advanced Vehicle Technology (Second Edition)*, pages 584 – 634. Butterworth-Heinemann, Oxford, second edition edition, 2002.