A Data-driven Region Generation Framework for Spatiotemporal Transportation Service Management

Liyue Chen[1,5], Jiangyi Fang[2], Zhe Yu[3], Yongxin Tong[4], Shaosheng Cao[3*], Leye Wang[1,5*]
[1] Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, China
[2] Huazhong University of Science and Technology, Wuhan, China
[3] DiDi Chuxing, Hangzhou, China
[4]SKLSDE Lab and IRI, Beihang University
[5] School of Computer Science, Peking University, Beijing, China
chenliyue2019@gmail.com,shelsoncao@didiglobal.com,leyewang@pku.edu.cn

ABSTRACT

MAUP (modifiable areal unit problem) is a fundamental problem for spatial data management and analysis. As an instantiation of MAUP in online transportation platforms, region generation (i.e., specifying the areal unit for service operations) is the first and vital step for supporting spatiotemporal transportation services such as ride-sharing and freight transport. Most existing region generation methods are manually specified (e.g., fixed-size grids), suffering from poor spatial semantic meaning and inflexibility to meet service operation requirements. In this paper, we propose ***RegionGen***, a data-driven region generation framework that can specify regions with key characteristics (e.g., good spatial semantic meaning and predictability) by *modeling region generation as a multi-objective optimization problem*. First, to obtain good spatial semantic meaning, *RegionGen* segments the whole city into atomic spatial elements based on road networks and obstacles (e.g., rivers). Then, it clusters the atomic spatial elements into regions by maximizing various operation characteristics, which is formulated as a multi-objective optimization problem. For this optimization problem, we propose a multi-objective co-optimization algorithm. Extensive experiments verify that *RegionGen* can generate more suitable regions than traditional methods for spatiotemporal service management.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; • **Applied computing** → **Transportation**.

KEYWORDS

Modifiable areal unit; spatial data management

*Corresponding author.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Liyue Chen, Jiangyi Fang, Zhe Yu, Yongxin Tong, Shaosheng Cao, Leye Wang. 2018. A Data-driven Region Generation Framework for Spatiotemporal Transportation Service Management. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 17 pages. https://doi.org/XXXXXXX.XXXXXXX

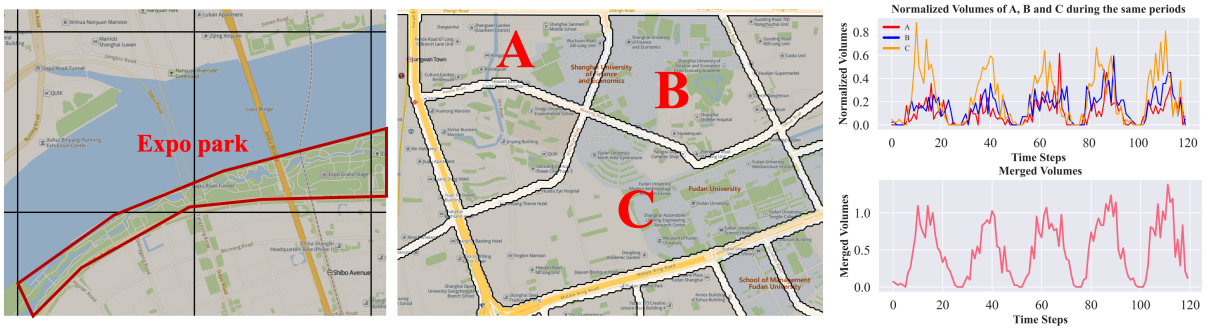
1 INTRODUCTION

The global transportation services market is expected to grow to over 7,200 billion and 10 trillion dollars in 2022 and 2026 over a compound annual growth rate of 9% according to the market report [49]. For online transportation platforms (e.g., Uber), *region is the fundamental operation areal unit for spatiotemporal data management.* For example, aiming to shorten passengers' waiting time, online transportation platforms first divide the whole city into several regions (e.g., $1km \times 1km$ grids [18, 68]) and dispatch idle drivers to hot areas based on the estimated demand for each region. Only with a well-managed set of regions, online transportation platforms can dispatch more orders to drivers and respond to passengers' ride-sharing requests more quickly.

In general, how to determine the region (i.e., areal unit) for spatial data management is one of the most important problems in geo-science, widely known as the *the modifiable areal unit problem* (MAUP) [45, 59], since different region specifications may lead to diverged analysis outcomes [8]. Despite the importance of the region specification, current industry practice is mostly ad-hoc and manually defined, e.g., *fixed shapes like grids or hexagons* [18, 29, 34, 68], and *street/administrative blocks* [39, 66, 69, 73, 74], without comprehensive assessment standards. This could incur uncontrollable and unanticipated effects on the operations of spatiotemporal services. As later shown in our experiments, compared to ad-hoc grid regions, an optimized region specification can reduce the spatiotemporal service demand prediction error by around 10%; this improvement is particularly noteworthy considering that a recent benchmark study [55] has shown that state-of-the-art deep models can improve prediction performance also by about 10% compared to classical methods (e.g., gradient boosted regression trees [32]). Hence, region specification is a crucial but largely under-investigated issue for spatiotemporal data management practice. Next, we briefly illustrate existing region specification methods and their pitfalls.

Typically, *fixed-shape* methods divide the whole city into numerous regions with the same shape, such as grids and hexagons, **in the absence of geographic semantic meaning**.[1] As demonstrated in

[1]Regions with good semantic meaning are bounded by roads [67].



(a) Examples of grid partition methods (b) Locations of Region A, B, and C (c) Volumes pattern in different regions

Figure 1: (a): Examples of grid partition methods that generate regions with poor spatial semantic meaning. The main green part is a park. (b) and (c): Generated by road segmentation methods, Region A, B, and C are with poor predictability, which may be improved by merging these three regions.

Fig. 1(a), the Shanghai Expo Park is an entire geographic entity, but it is separated into several grids, violating the semantics of urban spaces. In addition, crossing a river usually takes longer for both automobiles and pedestrians than crossing a road. However, the fact that the two sides of the river are divided into the same grid makes it inconvenient to operate online transportation services. For instance, when a driver is guided to such a grid to wait for ride-sharing requests, she/he would be doubtful whether she/he needs to go across the river or not.

Street/administrative-block methods use pre-defined spatial boundaries (i.e., roads and administrative boundary) for region generation. While these methods can keep good geographic semantic meaning compared to *fixed-shape* methods, they generally **lack the flexibility to well fit various operation requirements** of online transportation services. For instance, administrative blocks may be too coarse to enable fine-grained operations (especially for downtown areas with a large number of demands); meanwhile, road-segmented street blocks may be too small and thus most blocks hold nearly zero demands.

A practically key spatiotemporal data management question then emerges: **can we generate a good set of regions for online transportation platforms by meeting two goals: (i) with good geographic semantic meaning, and (ii) flexibly fitting various operation requirements of spatiotemporal services?**

To address the above problem, we propose and implement a unified region generation framework, called ***RegionGen***, which enables the adaptive formation of regions for various spatiotemporal transportation services, such as *taxi dispatching*, *freight transportation*, and *designated driver* services. The general process of *RegionGen* follows two steps. In the first step, we adapt road segmentation techniques [67] to extract semantically-meaningful atomic spatial units. While the atomic spatial units may be too small for operation, the second step merges nearby units into larger regions, which can be seen as a spatial clustering process.

In particular, our spatial clustering process needs to consider various realistic operation characteristics for spatiotemporal services, including generated regions' granularity, specificity, etc. [4, 22, 23, 32]. More specifically, we argue that the *temporal predictability* plays a key role in transportation service operations. If the regions are generated with higher predictability, future events of interests (e.g., the number of ride-sharing requests in next one hour) are easier to forecast and thus service operation decisions can be made more accurately (e.g., how many drivers need to be dispatched to a certain region). Fig. 1(b) and (c) illustrate a toy example to show how spatial clustering can help generate a high-predictability region for freight services. A, B, and C are three atomic spatial units generated by road segmentation; nevertheless, their temporal patterns on freight are unstable, as each spatial unit contains fewer data and are susceptible to random disturbance. Then, performing operation management directly based on these three spatial units will be difficult. Meanwhile, we find that these three spatial units all contain university campuses; then, by clustering them together for operation, we obtain a highly-predictable merged region whose temporal regularity is significantly increased.

In summary, our main contributions include:

- To the best of our knowledge, this is one of the pioneering efforts toward generating regions considering key operation characteristics (e.g., good spatial semantic meaning and predictability) in spatiotemporal service. This is also one of the first data-driven solutions to MAUP for spatiotemporal transportation service operations.
- *RegionGen* includes two main steps. First, to keep good spatial semantic meanings, *RegionGen* segments the whole city into atomic spatial elements based on road networks and obstacles (e.g., rivers). Second, it clusters the atomic spatial elements by maximizing the key operating characteristics with a multi-objective optimization process.
- We conduct extensive experiments on three spatiotemporal service datasets. Results verify that *RegionGen* can generate more suitable regions than traditional methods for spatiotemporal transportation service.

2 OPERATION CHARACTERISTICS ANALYSIS AND PROBLEM FORMULATION

As introduced in Sec. 1, there are two main challenges in designing such a region generation framework. The first is to guarantee the generated regions with good spatial semantic meaning. The second is to make the generated regions meet the operation requirements of spatiotemporal services. Our basic idea is to aggregate segmented fine-grained regions into large ones to acquire better

operating characteristics (e.g., predictability). It is worth noting that the aggregated regions are still bounded by roads and not cross obstacle entities, and thus still guarantee good spatial semantics.

Accordingly, we disassemble the region generation process in two steps. The first step is to generate many fine-grained regions with good spatial semantics by segmentation (in the rest of the paper, for clarity, the fine-grained regions are called **atomic spatial elements**). Then, the second step is to aggregate atomic spatial elements to obtain better operating characteristics. Especially, we first quantify several spatiotemporal services operating characteristics (Sec. 2.1). While there exist a variety of spatiotemporal services in practice, we have summarized three types of general operating characteristics (i.e., predictability, granularity, and specificity) that may benefit most services.

2.1 Quantifying Operation Characteristics

In most urban prediction applications (e.g., ride-sharing or dockless bike-sharing demand prediction), we usually expect regions with pleasant **predictability**, fine-grained spatial **granularity**, and high service **specificity**. *Predictability* refers to whether future events of interest for the service (e.g., ride-sharing demands) can be easily predicted. At the same time, many spatiotemporal services (e.g., traffic monitoring) hope that the spatial *granularity* can be fine-grained (e.g., the region size is not too large) to carry out precise management. Besides, high *specificity* indicates that the generated region closely matches the actual service area, i.e., the generated region has few redundant parts where (almost) no service is required (e.g., ride-sharing service operation regions may not need to cover mountain areas where cars cannot reach). We here quantify these general characteristics for the region generation process.

Predictability. As there exist various prediction models that can be adopted in practice [55], it is generally non-trivial to efficiently and precisely quantify the predictability of the spatiotemporal time-series data within regions. An intuitive way could be firstly fixing a prediction model and then measuring the prediction errors on test records (e.g., mean absolute error, Kaboudan metric [26]). Such measurements are highly model-dependent. That is, the error measurements obtained by one model cannot usually represent another model. As state-of-the-art spatiotemporal prediction models are mostly complex deep models [55], training these models to measure predictability for every possible region generation candidate would be too time-consuming and thus practically intractable.

We then propose to use *model-agnostic* measures to efficiently estimate regions' predictability without the need to repeatedly training deep models. Specifically, we select the auto-correlation function (ACF) as a fast proxy measurement for predictability. Suppose that $\{s_{1,i}, s_{2,i}, ..., s_{T,i}\}$ is the time series of region i, the ACF of region i after k slots delay is computed by:

$$\rho_i^k = \frac{T \cdot \sum_{t=k+1}^{T}(s_{t,i} - \bar{s_i})(s_{t-k,i} - \bar{s_i})}{(T-k) \cdot \sum_{t=1}^{T}(s_{t,i} - \bar{s_i})^2} \quad (1)$$

where $\bar{s_i}$ is the mean value of the time series in region i. ACF is often used to measure the periodicity of time series data. While high periodicity (e.g., daily periodicity) is one of the dominant indicators for accurate prediction [55], we deem that high periodicity may be highly correlated with low prediction errors of deep models. To investigate whether such correlations exist, we explore the relationship between the ACF after 24 hours delay (called ACF_daily) and prediction errors regarding a state-of-the-art deep forecasting model (i.e., *STMGCN* [18]) in Fig. 10 (Appendix A.1). The results verify that ACF_daily and prediction errors are highly correlated, especially when ACF_daily is large; that is, maximizing the ACF_daily would practically decrease the prediction errors significantly. Therefore, in this work, we adopt the ACF_daily indicator for efficiently measuring the predictability of regions.

Granularity. In real-world applications, to keep good spatial semantics, it is difficult to enforce the generated region with specific shapes and we typically measure the region size by the region area. More importantly, keep in mind that the greater the area, the more spatiotemporal data it contains, and the less random noise in the time series, the more regular the time series are, and the more predictable they are. Therefore, unlike predictability, we cannot require that the generated regions be as small as possible. As a result, we need to make a trade-off between good predictability and fine-grained spatial granularity. In practice, we typically meet the need for fine-grained spatial granularity by imposing the maximum region area constraint. Suppose that P_i is the geographic shape of region i, stored as a set of vector coordinates, the granularity of region i is quantified by its area:

$$ts_i = \text{Area}(P_i) \quad (2)$$

where $\text{Area}(\cdot)$ is the area of the 2D polygon calculation function, having been widely integrated into tools, e.g., ArcGIS.[2]

Specificity. For most spatiotemporal services, the regions for operation management do not always need to cover the entire target area (e.g., a city), as many sub-areas may not have the service requirements (e.g., ride-sharing services may not be required for mountain sub-areas where vehicles cannot access). To this end, the generated regions suitable for operation management would prefer to cover only those service-required sub-areas. We thus propose the *specificity* to measure the ratio of service-required sub-areas within the generated regions. Then, low-specificity regions mean that there exist a lot of redundant sub-areas that do not have service requirements and may negatively impact the operation efficiency. For example, when providing online ride-sharing services, if the service specificity of a region is low, the ride-sharing demands are mainly concentrated in only a few hotspots. When dispatching idle drivers to this region, a large number of drivers may influx to the same place, causing traffic congestion.

Particularly, we define the ratio of the service-required area (vs) to the total area of the region (ts) as the *specificity*, where the service-required area is counted according to historical service records. While spatiotemporal data records usually store spatial information in the format of points (e.g., latitude and longitude), we first convert these data points to Geohash [58] units for further area size calculation. Especially, we use 8-bit geohash (Fig. 12 in Appendix A.3 displays the geohash in the service-required area and the whole area) as the calculation unit to get an efficient and effective approximation to the area size. The service specificity of

[2] https://www.arcgis.com/

the region i is calculated as:

$$c_i = \frac{vs_i}{ts_i} \approx \frac{\#\text{ of historically-serviced geohash in region } i}{\#\text{ of total geohash in region } i} \tag{3}$$

2.2 Problem Formulation

2.2.1 Atomic Spatial Element Segmentation Problem. Given road networks and geographic obstacles (e.g., rivers), the segmentation problem aims to generate road segments bounded by roads and not overlapping with obstacles.

2.2.2 Atomic Spatial Element Clustering Problem. Given a graph G of N nodes (i.e., the atomic spatial elements $\mathcal{P}$), the adjacency matrix $A \in \mathbb{R}^{N\times N}$ (details in 3.3.1), spatiotemporal raster data $D \in \mathbb{R}^{T\times N}$ which is extracted from $\mathcal{P}$ and historical service records $\mathcal{D} = \{(l_i, t_i)\}$ (l_i and t_i are the locations and the timestamp that the service takes place), the number of time intervals T, maximum area constraint L, and the service-required area and total area $vs, ts \in \mathbb{R}^{N\times 1}$ of atomic spatial elements, we aim to cluster N atomic spatial elements to M clusters ($2 \le M < N$) by maximizing the average predictability and service specificity of clusters.

$$\text{Maximize } f_1(X) = \frac{1}{M}\sum_{j=1}^{M} \rho_j^{daily} \tag{4}$$

$$\text{Maximize } f_2(X) = \frac{1}{M}\sum_{j=1}^{M} \frac{\sum_{i=1}^{N} x_{i,j}\cdot vs_i}{\sum_{i=1}^{N} x_{i,j}\cdot ts_i} \tag{5}$$

$$\text{s.t.} \sum_{j=1}^{M} x_{i,j} = 1 \quad \forall i \in [N] \tag{6}$$

$$\sum_{i=1}^{N} x_{i,j} \ge 1 \quad \forall j \in [M] \tag{7}$$

$$\sum_{i=1}^{N} x_{i,j}\cdot ts_i \le L \quad \forall j \in [M] \tag{8}$$

$$x_{u,i} + x_{v,i} - \sum_{z\in\mathcal{S}} x_{z,i} \le 1 \quad \forall A_{u,v} = 0, \mathcal{S} \in \Gamma(u,v) \tag{9}$$

where $X \in \{0,1\}^{N\times M}$ is the binary clustering results. $x_{i,j} = 1$ denotes i^{th} atomic spatial element belong to j^{th} cluster. $S = D\times X \in \mathbb{R}^{T\times M}$ is the aggregated ST raster data. ρ_j^{daily} is the ACF of the time series in the j^{th} cluster after one day delay. Eq. 6 impose each atomic spatial element can only be assigned to at most one cluster. Ineq. 7 ensures that each cluster contains at least one atomic spatial element. The overall area of each aggregated cluster must be less than the specified maximum area (Ineq. 8). Ineq. 9 guarantees that every cluster induces a connected subgraph, where u, v are two non-adjacent nodes in G. A set $\mathcal{S} \subseteq V\backslash\{u,v\}$ is a (u,v)-separator if u and v belong to different components of $G - S$. We denote by $\Gamma(u,v)$ the collection of all minimal (u,v)-separators in G [44].

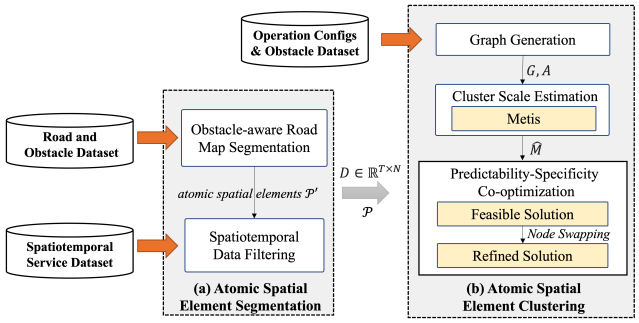


Figure 2: Overview of *RegionGen* framework.

3 METHOD

3.1 Framework Overview

The proposed region generation framework *RegionGen* consists of two core components, including the atomic spatial element segmentation and clustering (Fig. 2). The atomic spatial elements segmentation component first extracts atomic spatial elements with good spatial semantic meaning by using the proposed obstacle-aware road map segmentation techniques. Then, the spatiotemporal data filtering block removes the atomic spatial elements with less service data to reduce the scale of the subsequent clustering problem. The atomic spatial elements clustering component first represents the atomic element clustering problem in graph formats, where the nodes are the atomic elements. By combining domain knowledge, we establish the edge sets, which indicate that related nodes can be aggregated into the same cluster. The cluster scale estimation component provides a minimal cluster number that enables the aggregated clusters to meet the operation requirements (i.e., granularity in Eq. 8), allowing for adapting the well-researched graph partition approaches to address the clustering problem that requires the fixed partition number (i.e., clustering number) as inputs. Finally, the predictability-specificity co-optimization component generates regions by solving the atomic spatial element clustering problem.

3.2 Atomic Spatial Element Segmentation

3.2.1 Obstacle-aware Road Map Segmentation. Road segments provide us with more natural and semantic meaning than *fixed-shape* methods. In the real world, geographic entities (e.g., parks and residential areas) are bounded by roads and people live in these roads-segmented regions and POIs (Points of Interests) fall in these regions instead of the main roads [67]. Previous research [39, 66, 67, 74] adopt image-based road segmentation techniques that mainly consist of three morphological operators, namely dilation, thinning, and connected component labeling (CCL). Dilation is designed to remove some redundant road details for segmentation avoiding the small connected areas induced by these unnecessary details (e.g., the lanes of roads and the overpasses). The thinning operator aims to extract the skeleton of the dilated road segments while keeping the topology structure of the original image. The CCL operator finds the connected pixels with the same label in the image and eventually generates road segmentation.

Although these road segmentation techniques can generate regions bounded with roads and keep good spatial semantic meaning. We argue that the generated regions may still be inconvenient for

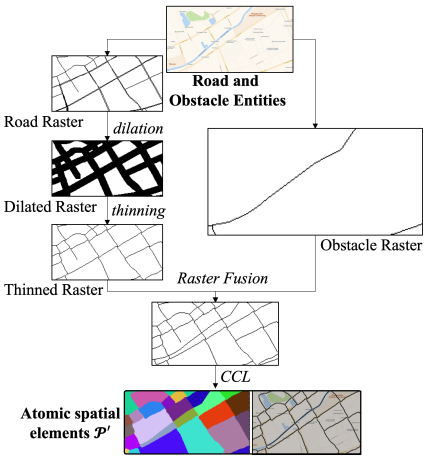


Figure 3: Obstacle-aware road map segmentation.

operating spatiotemporal services since the regions may cross obstacle entities (e.g., rivers). To overcome this shortage, we proposed the obstacle-aware road map segmentation method that incorporates obstacle entities into the road map segmentation process, enabling the road segmentation will not to stretch over obstacle entitles and thus be more suitable for operating services.

Fig. 3 shows the procedure of the obstacle-aware road map segmentation method and the main intermediate results of an example. The main improvement lies in the obstacle entities. Geographic obstacle data usually stores in vector form in spatial databases (e.g., OpenStreetMap) and we convert the vector-based obstacle data into binary images. Each pixel represents a grid cell ('1' denotes the obstacle segments, and '0' stands for the background). Then, we fuse the obstacle raster with thinned road raster (after dilation and thinning), so that the obstacles entities will also be the boundaries for segmentation. The generated regions are called **atomic spatiotemporal elements** $\mathcal{P}'$. Besides, we should use fine-grained level road data since we expect fine-grained spatial operation and the atomic spatial elements need to be as small as possible.

3.2.2 Spatiotemporal Data Filtering. The obstacle-aware road map segmentation method outputs $\mathcal{P}'$, containing N' atomic spatial elements. Due to the heterogeneous spatial data density, many spatial atomic elements hardly contain spatiotemporal service data and are unnecessary to store. Filtering them helps reduce the scale of subsequent clustering problems and obtain better spatial granularity for supporting spatiotemporal services. We impose restrictions on the atomic elements based on the service data, filter out atomic elements whose data amount (e.g., daily average demand) is smaller than α, and then get N main atomic spatial elements.

3.3 Atomic Spatial Element Clustering

3.3.1 Graph Generation. To solve the clustering problem for region generation (Sec. 2.2.2), we transform the problem into graph format so that the connected constraint (i.e. Eq. 9) is easily satisfied by adopting many well-studied techniques (e.g., connected graph partition). Every atomic spatial element can be regarded as a node in the graph and their edges denote their 'aggregatable' property. That is, the edges between two nodes represent that the predefined constraints (e.g., maximum area and geographic adjacent constraints) are still satisfied after merging these two connected nodes. In this section, we elaborate on building the edge sets of atomic spatial elements based on domain knowledge and geographic constraints.

Domain Knowledge. The atomic spatial elements may have the following properties, which indicate that the atomic elements cannot or unnecessarily aggregate with other atomic elements:

- *Oversize:* The road network may be scarce in some places (e.g., suburbs), and obstacle-aware road map segmentation may produce huge atomic elements whose area may exceed the maximum area specified by the service operator. At this time, aggregating the large atomic elements will generate an oversized region, which makes the operation difficult.
- *Already Predictable:* In urban hot spots, such as commercial areas, small atomic elements may contain a lot of spatiotemporal data, and they may be predictable even if not aggregated with other atomic elements. In this situation, it is not necessary to aggregate, and small regions can retain fine-grained spatial granularity.

Geographic Constraints. The 'aggregatable' nodes must meet the following two geographic constraints (denote as C), otherwise, the clustered regions are useless for spatiotemporal services:

- *Adjacent Constraint:* 'Aggregable' atomic elements should be geographically adjacent. In practice, we usually set a small threshold τ (e.g., 50 m), and when the shortest distance between the two atomic elements is less than τ, two atomic regions are defined as geographically adjacent.
- *Obstacle Constraint:* Regions with good spatial semantics do not cross obstacle entities (e.g., rivers). To ensure that the aggregated regions still keep good spatial semantics, if there is an obstacle between the two atomic elements, there will be no edge between these two atomic spatial elements.

Fig. 4 shows an example of the graph generation process of 7 atomic spatial elements. First, based on the ACF and area attributes, Node 1 and Node 3 will be separate nodes due to oversize area and high predictability (i.e., ACF). We then calculate the geographic distance for the remaining 5 nodes (Node 2, 4, 5, 6, 7). Among these 5 nodes, there is a river between Node 2 and the remaining nodes, so there are no edges between them (marked with red dotted lines). Next, the distances between Node 2 and Node 4, Node 4 and Node 6, etc., exceed the given threshold ($\tau = 50m$) and therefore do not have edges between them (marked with blue dotted lines). Node 4 and Node 5, Node 5 and Node 6, etc., are closely adjacent and do not cross the river, and thus, can be aggregated (marked with black lines). With the above process, the graph generation module eventually produces a graph G with atomic elements as nodes (V) and their adjacency matrix A (equivalent to the edge set E of G).

3.3.2 Cluster Scale Estimation. Another challenge for solving the clustering problem is to guarantee the solutions can satisfy the maximum area constraints (Ineq. 8). Note that the maximum area constraint is highly related to the cluster number (i.e., M). For example, for a 100 km^2 city, if the maximum area for each region is 5 km^2, we may require 20 regions to operate services; if the constraint changes to 10 km^2, we only need 10 regions. We need to estimate the clustering scale so that the clustered results can meet the maximum area constraints. A natural way to estimate

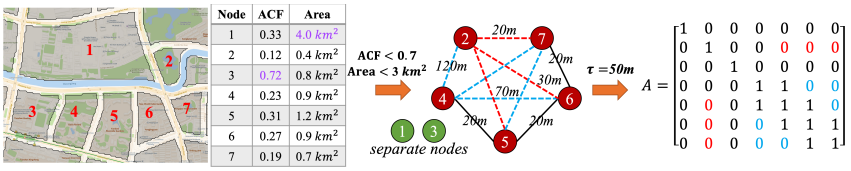


Figure 4: An illustrative example of the graph generation process. The nodes represent 7 atomic spatial elements and the adjacent matrix is calculated based on the geographic distance and the obstacle entities. The red and blue dotted lines denote that they can not be merged into the same cluster due to the river and far distance respectively.

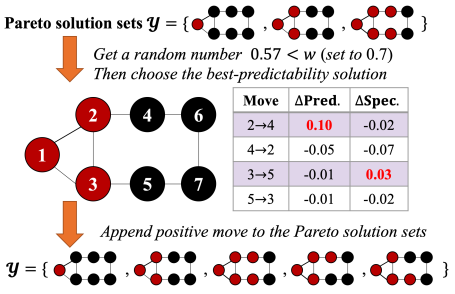


Figure 5: An example of the predictability-specificity co-optimization process during an iteration. A→B: append B to the cluster of A.

is gradually increasing M and checking whether the maximum area constraint is satisfying. For clarity, we first defined the ideal minimum cluster number as M^*, which denotes the cluster scale when the optimal solution of the clustering problem is obtained under the maximum area constraint. M^* satisfies:

$$\lceil \frac{\sum_{i=1}^{N} ts_i}{L} \rceil < M^* < N \tag{10}$$

where the low bound represents the results that the area of each cluster is L (recall that ts_i is the area of i^{th} atomic spatial element) and it scarcely occurs in real applications. Getting the value range of M^*, we find M^* by searching from the low bound to the upper bound. We denote $\widehat{M}$ as the minimum cluster scale — if we set $M < \widehat{M}$, we cannot obtain a feasible aggregation result satisfying the maximum area constraint.

Then, given a trial cluster scale, we need a *fast solver* to obtain a feasible aggregated solution to test whether the maximum area constraint is satisfying. The fast solver is different from the optimization solver (will be discussed in Sec. 3.3.3), as the optimization solvers are usually slow and inefficient for estimating $\widehat{M}$. Fig. 11 (Appendix A.2) reveals that, if a region has more data, it has more obvious daily regularity and corresponding better predictability. Hence we need to balance the data amount throughout the clusters to prevent any cluster from having too little data volume (resulting in poor predictability), as we would like to get better predictability for every cluster (Eq. 4). Existing balanced graph partition methods [1, 28, 50] are efficient enough to be the *fast solver* for the proposed clustering problem (a sparse graph whose nodes are less than 10k), as they can complete the partitioning of a graph with more than 10k nodes in 3 seconds while tolerating 5% unbalance [50]. For every trial cluster scale, we will check whether the current balanced graph partition results satisfy the maximum area constraint. Specifically, we use *Metis* [27], a popular multi-level graph partition software package, as the *fast solver*. We assign the node weight with the amount of spatiotemporal data and minimize the edge-cut (to isolate small degrees nodes) while tolerating 5% unbalance. We call this method *D-Balance*.

3.3.3 Predictability-Specificity Co-optimization. The above *fast-solver* can generate feasible but insufficiently superior solutions, and the specificity is not taken as the optimization objective. Hence, we design a heuristic predictability-specificity co-optimization algorithm that can iteratively refine multiple objectives, based on the famous node-swapping local search approaches (i.e., Fiduccia-Mattheyses [16]). We summarize the proposed algorithm in Algorithm 1 (Appendix B). Our algorithm first initializes Pareto solution sets $\mathcal{Y}$ with simple strategies (e.g., random growth or greedy). There are three initialization methods (details in Appendix E.1):

- *D-Balance* balances the data across cluster (details in Sec. 3.3.2).
- *Greedy* [28] grows clusters by choosing the maximum gain.
- *Fluid* [47] aggregates nodes by random propagation.

Then, at each iteration, it first selects a candidate solution from $\mathcal{Y}$ and iteratively improves it by moving boundary nodes that connect two clusters to obtain the positive gain (obtaining better predictability or specificity) while not violating the geographic constraint C.

The way of selecting the candidate solutions from Pareto solution sets $\mathcal{Y}$ will determine the final solution quality. We hold the Pareto optimal solutions rather than weighting them into a single objective (e.g., linear scalarization [43]). At the beginning of each iteration, we sample a random number $\in [0, 1]$ from uniform distribution and compare it with a predefined parameter $w \in [0, 1]$. If w is bigger, we choose the best-predictability solution (probability w) from $\mathcal{Y}$. Otherwise, we select the best-specificity solution (probability 1-w). That is, w represents the preference (probability) of optimizing the predictability objective. As the example shown in Fig. 5, we first choose the candidate solution with probability $w = 0.7$. Then the selected best-predictability solution will serve as the starting solution for the refinement. We record all positive gain movements and append them into $\mathcal{Y}$ for the next iteration. Note that, in one iteration, even if we choose the best-predictability solution for refinement, we still record the better-specificity solutions during refinement. The algorithm stops when no more positive gain can obtain or achieve the max iterations.

Time Complexity. The refinement process improves solutions by swapping the nodes having edges, and we will try every pair of nodes in the worst case. Since we limit the max iteration number to n, the time complexity of the co-optimization algorithm is $O(n|E|)$, where E is the edge set of G.

4 IMPLEMENTATION AND DEPLOYMENT

As illustrated in Fig. 6, our system mainly has two parts, i.e., offline periodical region optimizing and online region query for the transportation services. In the offline optimizing part, we adopt the

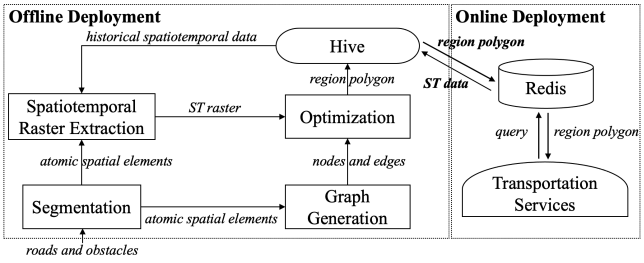


Figure 6: The architecture of *RegionGen* system.

'T+1' mode to update the generated regions, which means the regions will be optimized and uploaded to Hive [3] in an offline manner on a daily basis and will be used for downstream spatiotemporal transportation services for the next day. In the online part, Redis[4], an online real-time database, daily updates the region polygons from Hive and respond to online queries from online transportation services. For example, when providing demand heatmap services, region polygons are queried for visualization.

The offline part of *RegionGen* is capable of optimizing fine-grained spatial atomic elements on a daily basis while the online transportation service only needs to look up the region polygons from Redis and the response time is around 100ms. The current run time of *RegionGen* is around 2 hours (Sec. 5.4) with serial processing, which is already enough for 'T+1' daily updates. Meanwhile, it can be accelerated by parallel processing. In each iteration, the co-optimizing algorithm in Sec. 3.3.3 refines solutions by moving all boundary nodes, which can be parallelized (i.e., each processor deals with a part of nodes). In practice, our *RegionGen* system has deployed to support online real-world services, such as demand heatmap visualization.

5 EVALUATION

5.1 Datasets, Baselines, and Experiment Settings

We collect three transportation service datasets (designated driver, freight transport, and taxi demand) as well as the geographic entities (roads and obstacles). Dataset details are in Appendix C. We implement widely-adopted manually-specified region generation methods (*Grid*, *Hexagon*) and data-driven baselines (*DBSCAN* [10], *GSC* [32], *GCSC* [4]). Our experiment platform is a server with 10 CPU cores (Intel Xeon CPU E5-2630), and 45 GB RAM. More baseline and experimental setting details are in Appendix D.

5.2 Evaluating with Spatiotemporal Prediction

To evaluate the effectiveness of *RegionGen* for accurately operating spatiotemporal services, we conduct demand predictions on three spatiotemporal service datasets, which are the basic capabilities of various downstream tasks, such as scheduling idle drivers. Specifically, we predict the demand in the next hour for all datasets.

5.3 Evaluation Metric

We evaluate the quality of the generated region by two operational characteristics, namely **ACF_daily** and service **specificity**. Besides, popular metrics (including RMSE and MAE) are not feasible to directly evaluate the prediction performance, since the prediction ground truths of various region generation methods are different. The Mean Absolute Percentage Error (**MAPE**) measures the relative errors and thus different predictive objects are comparable. However, different regions contain varying amounts of service data, to make a fair comparison, we ensure the amount of service data of different regions is approximately equal (i.e., demand recall). Specifically, we filter out the regions whose average daily demand is less than 1, and get recalls of 97%, 98%, and 99.8%, respectively, in the designated driver, freight transport, and taxi demand datasets.

5.4 Results and Analysis

5.4.1 Main results. In Table 1, we report the ACF_daily, Specificity, and MAPE@Recall on the designated driver and freight transport dataset under the same clustering scale. *RegionGen* gets Pareto optimal solutions (w is 0.7) and we report two solutions best in the ACF_daily and specificity metrics respectively. In Table 2, we report the results of ACF_daily and MAPE@Recall on the taxi dataset, since the latitude and longitude of the original data are aggregated into the census tracts, making the 'specificity' metric inapplicable. Table 1 shows that *RegionGen* achieves better ACF_daily than the baselines and gets corresponding lower MAPE@Recall. Especially, *RegionGen (Best ACF)* consistently outperforms baselines in terms of MAPE@Recall, decreasing 3.2% and 3.3% than *Grid* in the designated driver and freight transport dataset. The above observations demonstrate the generated regions with better predictability support operating more accurate services. They also provide us with new insight that, besides predictive models, the prediction performance can be significantly improved by generating regions with better predictability. The results on the taxi dataset in Table 2 are similar. *RegionGen* consistently gets the best ACF_daily and prediction accuracy. In addition, we record the run time of *RegionGen*. For one city, *RegionGen* can finish region generation in two hours, which is enough for real-life deployment and usage (detailed discussion in Sec. 4).

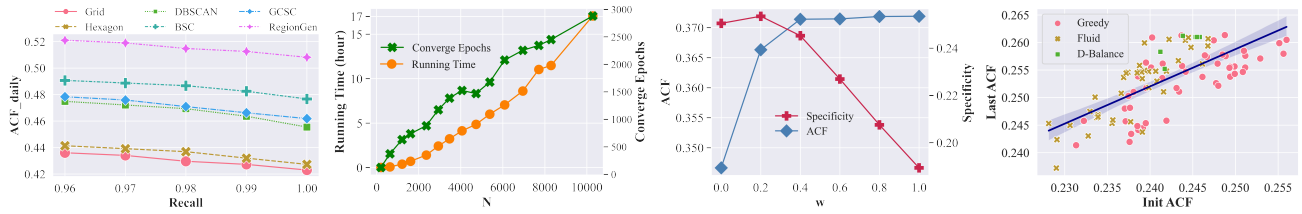
5.4.2 Robustness analysis on open datasets. The main results are conducted on two private spatiotemporal service datasets. To help reproduce our results, we also experiment on an open dataset, Chicago taxi demand. Moreover, we conduct spatiotemporal prediction based on various state-of-the-art models, including *STMGCN* [18], *GraphWaveNet* [61] and *STMeta* [55], as these three models have performed competitively in a recent benchmark study [55]. Results in Table 2 show that *RegionGen* outperforms baselines consistently. Despite small differences in the prediction accuracy of the three models, MAPE is highly correlated to ACF_daily. For instance, *RegionGen* obtains the highest ACF_daily, and achieves the lowest MAPE consistently under three models. This confirms that our choice of ACF_daily as a measurement for the predictability of regions is acceptable and effective.

5.4.3 Robustness analysis under different recall settings. To see whether *RegionGen* is robust under different recall settings, we unceasingly remove the tailed atomic spatial elements that are with the fewest data samples. Fig. 7(a) show how ACF_daily changes by varying the recall. We observe that *RegionGen* (marked with

[3] https://hive.apache.org/
[4] https://redis.io/

Table 1: Results on designated driver and freight transport dataset. M is the number of regions. The best two results are highlighted (*best* is in bold and *italic*, second-best is in bold). The MAPE@Recall results are given by *STMGCN* [18]. The specificity of *DBSCAN* is not applicable (N/A) since it is a point clustering method.

	Designated Driver (M=913)				Freight Transport (M=724)			
	ACF_daily	Specificity	MAPE@97%	Run Time	ACF_daily	Specificity	MAPE@98%	Run Time
Baselines								
Grid	0.4186	0.3607	0.3647	<1s	**0.3596**	0.1701	**0.3634**	<1s
Hexagon	0.4307	0.3781	0.3615	<1s	0.3579	0.1853	0.3657	<1s
DBSCAN	0.3998	N/A	0.3711	~7mins	0.3189	N/A	0.4092	~4mins
BSC	0.4421	0.3791	0.3580	~4mins	0.3378	0.1879	0.3852	~5mins
GCSC	0.4240	0.3615	0.3639	~5mins	0.3295	0.1812	0.3966	~3mins
RegionGen (best specificity)	**0.4577**	***0.3902***	**0.3501**	~100mins	0.3497	***0.2535***	0.3698	~50mins
RegionGen (best ACF)	***0.4740***	**0.3851**	***0.3321***	~100mins	***0.3719***	**0.2502**	***0.3305***	~50mins



(a) (b) (c) (d)

Figure 7: (a) ACF_daily vs. recall; (b) Speed vs. the Scalability; (c) Performace vs. Parameter w; (d) Effect of initialization methods

Table 2: Results on taxi demand dataset. M is the number of regions. The best two results are highlighted (*best* is in bold and *italic*, second-best is in bold). The original dataset stores the location at the census-tract level, so "Specificity" cannot be computed; *RegionGen* is set to optimize ACF only (w=1).

	Taxi Demand (M=95)			
	ACF_daily	MAPE@99.8%		
		STMGCN	*STMeta*	*GraphWaveNet*
Baselines				
Grid	0.3018	0.4152	0.4050	0.3904
Hexagon	0.3076	0.4094	0.4016	0.3855
DBSCAN	0.3323	0.4011	0.3921	0.3776
BSC	**0.3691**	**0.3526**	**0.3456**	**0.3308**
GCSC	0.3402	0.3769	0.3674	0.3494
RegionGen	***0.3841***	***0.3409***	***0.3295***	***0.3123***

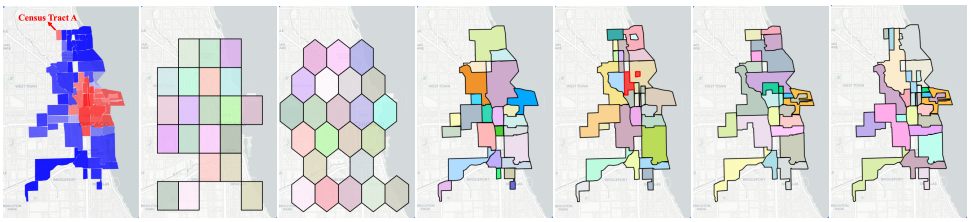
red lines) consistently achieves the best ACF_daily, demonstrating its robustness. Among baselines, with the recall decreasing, ACF_daily of *DBSCAN* increases the fastest (green lines). It shows that *DBSCAN* has a relatively obvious long-tail phenomenon; that is, the ACF_daily of the tailed elements is much lower than the other elements, which may incur inconvenience for operating services upon these tailed regions. Besides, fixed-shape methods, *Grid* and *Hexagon* (blue lines) perform consistently the worst. Note that fixed-shape regions are still popular for spatiotemporal service management in practice, but these results again point out their weakness. Hence, it would be expected that new region generation technologies, such as *RegionGen*, will significantly advance the field.

5.4.4 Analysis of scalability. To analyze the scalability of *RegionGen*, we conduct experiments on different scales (i.e., the number of atomic elements N). Fig. 7(b) shows our results. We explored the change of *RegionGen* with the scale by dividing the target area into different numbers of atomic elements (from 100 to 10,000).[5] We observe that as the scale increases, the running time and the converge epochs of the algorithm also increase synchronously. It is worth noting that 10,000 atomic elements can already support fine-grained spatiotemporal service in a metropolis like Shanghai (i.e., each atomic element covers around 0.01 km^2); with 10,000 atomic elements, *RegionGen* takes 17 hours, which still satisfies the need for the 'T+1' update mode in our realistic deployment (Sec. 4). This demonstrates that *RegionGen* is capable of optimizing very fine-grained spatial atomic elements.

5.4.5 Effect of w. In Fig. 7(c), we display the ACF metric of the best-predictability solution and the Specificity metric of the best-specificity solutions from different Pareto solution sets obtained by changing w, the probability of optimizing the predictability objective. We observe that: (i) with the increase of w, the ACF metric gradually increases and the specificity metric decreases; (ii) when w is set to 0.4, the w-hold mechanism prefers to optimize specificity, but still achieves solutions with reasonable predictability. Hence, in practice, we may obtain a solution with both good predictability and high specificity by setting an appropriate w.

5.4.6 Effect of different initialization methods. In Fig. 7(d), we compare different initialization methods including *D-Balance*, *Greedy*, and *Fluid* by generating initial solutions with 30 different random seeds for each method. We record the initial ACF (called Init ACF) and the ACF when the algorithm stops (called Last ACF). We observe that (i) the final solutions given by *Greedy* and *Fluid* are quite different with diverse random seeds (the difference in terms of

[5]In this experiment, we choose grids as the atomic spatial elements since they easily adapt to different granularity.



(a) Heatmap (b) Grid (c) Hexagon (d) DBSCAN (e) BSC (f) GCSC (g) RegionGen

Figure 8: Region clusters in Chicago generated by the baselines and RegionGen.

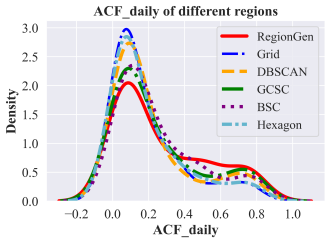


Figure 9: ACF distribution.

ACF exceeds 0.02); (ii) the Init ACF is linearly related to the last ACF, which inspires us to choose an initial solution with better quality for further optimization; (iii) The Last ACF of *D-Balance* solutions are better than *Greedy* and *Fluid* (closer to the upper area), demonstrating that it is more capable and more robust to be the initialization method for generating high-quality solutions.

5.5 Case Study

We visualize the generated regions of baselines and *RegionGen* in the downtown area of Chicago on the taxi demand dataset. We filter out regions with few historical service data for all region generation methods and each color represents a region. Fig. 8(a) displays the demand heatmap and red indicates high-density areas. In Fig. 8(b) and Fig. 8(c), *Grid* and *Hexagon* generate regions with poor spatial semantics, while hotspots and cold areas are of the same spatial granularity. Based on census tracts, in Fig. 8(d), regions clustered by *DBSCAN* are with good spatial semantic meaning. *DBSCAN* prefers to aggregate more census tracts in the high-density area (i.e., hotspots with many data points), which offers excellent predictability but low spatial granularity. At the same time, in cold areas with few data points, *DBSCAN* preserves a single census tract (good spatial granularity) but with bad predictability. In Fig. 8(e), the aggregate regions by *BSC* will not be oversize like *DBSCAN* since the first partition stage in *BSC* makes all aggregated regions geographically close. However, the nearby census tracts may not be strictly adjacent, and nonadjacent census tracts with similar demand matrices will also be aggregated (marked with red boxes), resulting in inappropriate clustering results. In Fig. 8(f), despite generating spatial continuous regions and obtaining sufficient adaptive granularity (small regions in hotspots and big regions in cold areas), *GCSC* may still fall short of successfully optimizing the predictability. For example, census tract A is predictable (with much data to support clear daily regularity), yet *GCSC* continues to aggregate it with other census tracts. In Fig. 8(g), *RegionGen* gets reasonable adaptive granularity and good predictability (small regions in hotspots and big regions in cold areas). Specifically, *RegionGen* takes census tract A (already predictable) alone as a cluster, demonstrating that it optimizes the predictability well.

In Fig. 9, we explore ACF_daily distribution of different regions on baselines and *RegionGen*. For *RegionGen* (red line), there are fewer regions with low ACF_daily than all baselines, while having more regions with larger ACF_daily. Therefore, *RegionGen* obtains better predictability by balancing the spatiotemporal data over different regions. That is, it prefers to cluster fewer atomic spatial elements in hotspots and more in cold areas.

6 RELATED WORK

With the wide adoption of GPS-equipped devices and the great success in machine learning models, massive historical spatiotemporal data (e.g., GPS trajectory data [30]) has been widely used to support intelligent transportation services, including traffic prediction [7, 9, 14, 19, 24, 31, 35, 48, 51, 52, 60], travel time estimation [2, 11, 12, 54, 64], transportation route recommendation [36–38], trajectory similarity computing and outlier detection [13, 21, 41, 62], bus route planning [17, 56, 57], outdoor advertising [71], and crowdsourcing [5, 53]. The transportation services may be operated upon specified areal units, e.g. by fixed-size grids [6, 15, 20, 25, 33, 46, 63, 65, 72]. Existing transportation services may benefit from our region generation framework. For example, for spatial crowdsourcing pricing applications (e.g., food delivery services), previous research demonstrated that spatial crowdsourcing needs to price for multiple local markets based on the spatiotemporal distributions of tasks and workers than seek a unified optimal price for a single global market [53]. The regions created by our framework are more suitable for estimating the spatiotemporal distribution of works (e.g., making the prediction of the supply of workers more accurate) and thus the pricing strategies are easier to give than grids [5, 53]. Moreover, with better spatial semantic meaning, our regions may support further analysis correlating with regions' functionality.

7 CONCLUSION

In this paper, we present a unified data-driven region generation framework, called *RegionGen*, which can flexibly adapt to various operation requirements of spatiotemporal services while keeping spatial semantic meaning. To keep the good spatial semantic meaning, *RegionGen* first segments the whole city into atomic spatial elements based on the fine-grained road networks and obstacle entities (e.g., rivers). Then, it aggregates the atomic spatial elements by maximizing key operating characteristics such as predictability and specificity. Extensive experiments have been conducted in three transportation datasets including two industrial datasets and an open dataset. The results demonstrate that *RegionGen* can generate regions with better operating characteristics (including spatial semantic meaning, predictability, and specificity) compared to other region generation baselines under the same granularity. Moreover, we conduct demand prediction services upon the generated regions, and *RegionGen* still achieves the best performance, verifying its effectiveness. As a pioneering study toward the adaptive region generation problem, we expect that our research can benefit online transportation platforms to provide intelligent and satisfactory transportation services.

REFERENCES

[1] Konstantin Andreev and Harald Räcke. 2004. Balanced graph partitioning. In *Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures*. 120–124.

[2] Richard Barnes, Senaka Buthpitiya, James Cook, Alex Fabrikant, Andrew Tomkins, and Fangzhou Xu. 2020. BusTr: Predicting Bus Travel Times from Real-Time Traffic. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 3243–3251. https://doi.org/10.1145/3394486.3403376

[3] Aydın Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. 2016. *Recent Advances in Graph Partitioning*. Springer International Publishing, Cham, 117–158.

[4] Longbiao Chen, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi-Mai-Trang Nguyen, and Jérémie Jakubowicz. 2016. Dynamic Cluster-Based over-Demand Prediction in Bike Sharing Systems. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 841–852. https://doi.org/10.1145/2971648.2971652

[5] Peng Cheng, Xun Jian, and Lei Chen. 2018. An Experimental Evaluation of Task Assignment in Spatial Crowdsourcing. *Proc. VLDB Endow.* 11, 11 (jul 2018), 1428–1440. https://doi.org/10.14778/3236187.3236196

[6] Teddy Cunningham, Graham Cormode, Hakan Ferhatosmanoglu, and Divesh Srivastava. 2021. Real-World Trajectory Sharing with Local Differential Privacy. *Proc. VLDB Endow.* 14, 11 (oct 2021), 2283–2295. https://doi.org/10.14778/3476249.3476280

[7] Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. 2020. Hybrid Spatio-Temporal Graph Convolutional Network: Improving Traffic Prediction with Navigation Data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 3074–3082. https://doi.org/10.1145/3394486.3403358

[8] Sidgley Camargo de Andrade, Camilo Restrepo-Estrada, Luiz Henrique Nunes, Carlos Augusto Morales Rodriguez, Júlio Cézar Estrella, Alexandre Cláudio Botazzo Delbem, and João Porto de Albuquerque. 2021. A multicriteria optimization framework for the definition of the spatial granularity of urban social media analytics. *International Journal of Geographical Information Science* 35, 1 (2021), 43–62.

[9] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang. 2021. ST-Norm: Spatial and Temporal Normalization for Multi-Variate Time Series Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 269–278. https://doi.org/10.1145/3447548.3467330

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Portland, Oregon) *(KDD'96)*. AAAI Press, 226–231.

[11] Xiaomin Fang, Jizhou Huang, Fan Wang, Lihang Liu, Yibo Sun, and Haifeng Wang. 2021. SSML: Self-Supervised Meta-Learner for En Route Travel Time Estimation at Baidu Maps. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 2840–2848. https://doi.org/10.1145/3447548.3467060

[12] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. 2020. ConSTGAT: Contextual Spatial-Temporal Graph Attention Network for Travel Time Estimation at Baidu Maps. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2697–2705. https://doi.org/10.1145/3394486.3403320

[13] Ziquan Fang, Yuntao Du, Xinjun Zhu, Danlei Hu, Lu Chen, Yunjun Gao, and Christian S. Jensen. 2022. Spatio-Temporal Trajectory Similarity Learning in Road Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 347–356. https://doi.org/10.1145/3534678.3539375

[14] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 364–373. https://doi.org/10.1145/3447548.3467430

[15] Ziquan Fang, Lu Pan, Lu Chen, Yuntao Du, and Yunjun Gao. 2021. MDTP: A Multi-Source Deep Traffic Prediction Framework over Spatio-Temporal Trajectory Data. *Proc. VLDB Endow.* 14, 8 (oct 2021), 1289–1297. https://doi.org/10.14778/3457390.3457394

[16] C.M. Fiduccia and R.M. Mattheyses. 1982. A Linear-Time Heuristic for Improving Network Partitions. In *19th Design Automation Conference*. 175–181. https://doi.org/10.1109/DAC.1982.1585498

[17] Tao-yang Fu and Wang-Chien Lee. 2021. ProgRPGAN: Progressive GAN for Route Planning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 393–403. https://doi.org/10.1145/3447548.3467406

[18] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (2019), 3656–3663.

[19] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. 2021. Dynamic and Multi-Faceted Spatio-Temporal Deep Learning for Traffic Speed Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 547–555. https://doi.org/10.1145/3447548.3467275

[20] Peng Han, Jin Wang, Di Yao, Shuo Shang, and Xiangliang Zhang. 2021. A Graph-Based Approach for Trajectory Similarity Computation in Spatial Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 556–564. https://doi.org/10.1145/3447548.3467337

[21] Xiaolin Han, Reynold Cheng, Chenhao Ma, and Tobias Grubenmann. 2022. DeepTEA: Effective and Efficient Online Time-Dependent Trajectory Outlier Detection. *Proc. VLDB Endow.* 15, 7 (jun 2022), 1493–1505. https://doi.org/10.14778/3523210.3523225

[22] Qiao-Chu He, Tiantian Nie, Yun Yang, and Max Shen. 2019. Beyond Rebalancing: Crowd-Sourcing and Geo-Fencing for Shared-Mobility Systems. *SSRN Electronic Journal* (01 2019). https://doi.org/10.2139/ssrn.3293022

[23] Qi Hu, Lingfeng Ming, Chengdong Tong, and Bolong Zheng. 2019. An Effective Partitioning Approach for Competitive Spatial-Temporal Searching (GIS Cup). In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Chicago, IL, USA) *(SIGSPATIAL '19)*. Association for Computing Machinery, New York, NY, USA, 620–623. https://doi.org/10.1145/3347146.3363349

[24] Bo Hui, Da Yan, Haiquan Chen, and Wei-Shinn Ku. 2021. TrajNet: A Trajectory-Based Deep Learning Model for Traffic Prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 716–724. https://doi.org/10.1145/3447548.3467236

[25] Yilun Jin, Kai Chen, and Qiang Yang. 2022. Selective Cross-City Transfer Learning for Traffic Prediction via Source City Region Re-Weighting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 731–741. https://doi.org/10.1145/3534678.3539250

[26] M. A. Kaboudan. 1999. A measure of time series' predictability using genetic programming applied to stock returns. *Journal of Forecasting* 18, 5 (1999), 345–357.

[27] George Karypis. 1998. A Software Package for Partitioning Unstructured Graphs , Partitioning Meshes , and Computing Fill-Reducing Orderings of Sparse Matrices Version 5 . 0.

[28] George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing* 20, 1 (1998), 359–392.

[29] Jintao Ke, Hongyu Zheng, Hai Yang, and Xiqun (Michael) Chen. 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies* 85 (2017), 591–608.

[30] Hai Lan, Jiong Xie, Zhifeng Bao, Feifei Li, Wei Tian, Fang Wang, Sheng Wang, and Ailin Zhang. 2022. VRE: A Versatile, Robust, and Economical Trajectory Data System. *Proc. VLDB Endow.* 15, 12 (sep 2022), 3398–3410. https://doi.org/10.14778/3554821.3554831

[31] Xiaoliang Lei, Hao Mei, Bin Shi, and Hua Wei. 2022. Modeling Network-Level Traffic Flow Transitions on Sparse Data. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 835–845. https://doi.org/10.1145/3534678.3539236

[32] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. 2015. Traffic Prediction in a Bike-Sharing System. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Seattle, Washington) *(SIGSPATIAL '15)*. Association for Computing Machinery, New York, NY, USA, Article 33, 10 pages. https://doi.org/10.1145/2820783.2820837

[33] Haoxing Lin, Rufan Bai, Weijia Jia, Xinyu Yang, and Yongjian You. 2020. Preserving Dynamic Attention for Long-Term Spatial-Temporal Prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 36–46. https://doi.org/10.1145/3394486.3403046

[34] Shuai Ling, Zhe Yu, Shaosheng Cao, Haipeng Zhang, and Simon Hu. 2022. STHAN: Transportation Demand Forecasting with Compound Spatio-Temporal Relationships. *ACM Trans. Knowl. Discov. Data* (oct 2022). https://doi.org/10.1145/3565578

[35] Dachuan Liu, Jin Wang, Shuo Shang, and Peng Han. 2022. MSDR: Multi-Step Dependency Relation Networks for Spatial Temporal Forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1042–1050. https://doi.org/10.1145/3534678.3539397

[36] Hao Liu, Jindong Han, Yanjie Fu, Jingbo Zhou, Xinjiang Lu, and Hui Xiong. 2021. Multi-Modal Transportation Recommendation with Unified Route Representation Learning. *Proc. VLDB Endow.* 14, 3 (dec 2021), 342–350. https://doi.org/10.14778/3430915.3430924

[37] Hao Liu, Ying Li, Yanjie Fu, Huaibo Mei, Jingbo Zhou, Xu Ma, and Hui Xiong. 2020. Polestar: An Intelligent, Efficient and National-Wide Public Transportation Routing Engine. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2321–2329. https://doi.org/10.1145/3394486.3403281

[38] Hao Liu, Yongxin Tong, Panpan Zhang, Xinjiang Lu, Jianguo Duan, and Hui Xiong. 2019. Hydra: A Personalized and Context-Aware Multi-Modal Transportation Recommendation System. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2314–2324. https://doi.org/10.1145/3292500.3330660

[39] Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xie Xing. 2011. Discovering Spatio-Temporal Causal Interactions in Traffic Data Streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Diego, California, USA) *(KDD '11)*. Association for Computing Machinery, New York, NY, USA, 1010–1018.

[40] Xuemei Liu, James Biagioni, Jakob Eriksson, Yin Wang, George Forman, and Yanmin Zhu. 2012. Mining Large-Scale, Sparse GPS Traces for Map Inference: Comparison of Approaches. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Beijing, China) *(KDD '12)*. Association for Computing Machinery, New York, NY, USA, 669–677. https://doi.org/10.1145/2339530.2339637

[41] Yiding Liu, Kaiqi Zhao, Gao Cong, and Zhifeng Bao. 2020. Online Anomalous Trajectory Detection with Deep Generative Sequence Modeling. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. 949–960. https://doi.org/10.1109/ICDE48307.2020.00087

[42] Jens Maue and Peter Sanders. 2007. Engineering Algorithms for Approximate Weighted Matching. In *Proceedings of the 6th International Conference on Experimental Algorithms* (Rome, Italy) *(WEA'07)*. Springer-Verlag, Berlin, Heidelberg, 242–255.

[43] Kaisa Miettinen. 2012. *Nonlinear multiobjective optimization*. Vol. 12. Springer Science & Business Media.

[44] Flávio K. Miyazawa, Phablo F.S. Moura, Matheus J. Ota, and Yoshiko Wakabayashi. 2021. Partitioning a graph into balanced connected classes: Formulations, separation and experiments. *European Journal of Operational Research* 293, 3 (2021), 826–836. https://doi.org/10.1016/j.ejor.2020.12.059

[45] Stan Openshaw. 1981. The modifiable areal unit problem. *Quantitative geography: A British view* (1981), 60–69.

[46] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban Traffic Prediction from Spatio-Temporal Data Using Deep Meta Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 1720–1730. https://doi.org/10.1145/3292500.3330884

[47] Ferran Parés, Dario Garcia Gasulla, Armand Vilalta, Jonatan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. 2017. Fluid communities: A competitive, scalable and diverse community detection algorithm. In *International conference on complex networks and their applications*. Springer, 229–240.

[48] Huiling qin, Xianyuan Zhan, Yuanxun Li, Xiaodu Yang, and Yu Zheng. 2021. Network-Wide Traffic States Imputation Using Self-Interested Coalitional Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 1370–1378. https://doi.org/10.1145/3447548.3467424

[49] Research and Markets. 2022. *Transport Services Global Market Report 2022: By Purpose, By Destination, By End-Use Industry*. https://www.researchandmarkets.com/reports/5568014 Accessed on November 7, 2022.

[50] Peter Sanders and Christian Schulz. 2013. Think locally, act globally: Highly balanced graph partitioning. In *International Symposium on Experimental Algorithms*. Springer, 164–175.

[51] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. Pre-Training Enhanced Spatial-Temporal Graph Neural Network for Multivariate Time Series Forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1567–1577. https://doi.org/10.1145/3534678.3539396

[52] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S. Jensen. 2022. Decoupled Dynamic Spatial-Temporal Graph Neural Network for Traffic Forecasting. *Proc. VLDB Endow.* 15, 11 (sep 2022), 2733–2746. https://doi.org/10.14778/3551793.3551827

[53] Yongxin Tong, Libin Wang, Zimu Zhou, Lei Chen, Bowen Du, and Jieping Ye. 2018. Dynamic Pricing in Spatial Crowdsourcing: A Matching-Based Approach. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 773–788. https://doi.org/10.1145/3183713.3196929

[54] Luan Tran, Min Y. Mun, Matthew Lim, Jonah Yamato, Nathan Huh, and Cyrus Shahabi. 2020. DeepTRANS: A Deep Learning System for Public Bus Travel Time Estimation Using Traffic Forecasting. *Proc. VLDB Endow.* 13, 12 (sep 2020), 2957–2960. https://doi.org/10.14778/3415478.3415518

[55] Leye Wang, Di Chai, Xuanzhe Liu, Liyue Chen, and Kai Chen. 2021. Exploring the Generalizability of Spatio-Temporal Traffic Prediction: Meta-Modeling and an Analytic Framework. *IEEE Transactions on Knowledge and Data Engineering* (2021), 1–1. https://doi.org/10.1109/TKDE.2021.3130762

[56] Sheng Wang, Zhifeng Bao, J. Shane Culpepper, Timos Sellis, and Xiaolin Qin. 2019. Fast Large-Scale Trajectory Clustering. *Proc. VLDB Endow.* 13, 1 (sep 2019), 29–42. https://doi.org/10.14778/3357377.3357380

[57] Sheng Wang, Yuan Sun, Christopher Musco, and Zhifeng Bao. 2021. Public Transport Planning: When Transit Network Connectivity Meets Commuting Demand. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 1906–1919. https://doi.org/10.1145/3448016.3457247

[58] Wikipedia. 2009. *Geohash*. https://en.wikipedia.org/wiki/Geohash Accessed on January 11, 2023.

[59] David WS Wong. 2004. The modifiable areal unit problem (MAUP). In *World-Minds: geographical perspectives on 100 problems*. Springer, 571–575.

[60] Dongxia Wu, Liyao Gao, Matteo Chinazzi, Xinyue Xiong, Alessandro Vespignani, Yi-An Ma, and Rose Yu. 2021. Quantifying Uncertainty in Deep Spatiotemporal Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 1841–1851. https://doi.org/10.1145/3447548.3467325

[61] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph Wavenet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (Macao, China) *(IJCAI'19)*. AAAI Press, 1907–1913.

[62] Di Yao, Haonan Hu, Lun Du, Gao Cong, Shi Han, and Jingping Bi. 2022. TrajGAT: A Graph-Based Long-Term Dependency Modeling Approach for Trajectory Similarity Computation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 2275–2285. https://doi.org/10.1145/3534678.3539358

[63] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, Xinran Tong, and Hui Xiong. 2019. Co-Prediction of Multiple Transportation Demands Based on Deep Spatio-Temporal Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 305–313. https://doi.org/10.1145/3292500.3330887

[64] Haitao Yuan, Guoliang Li, Zhifeng Bao, and Ling Feng. 2020. Effective Travel Time Estimation: When Historical Trajectories over Road Networks Matter. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 2135–2149. https://doi.org/10.1145/3318464.3389771

[65] Haitao Yuan, Guoliang Li, Zhifeng Bao, and Ling Feng. 2021. An Effective Joint Prediction Model for Travel Demands and Traffic Flows. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. 348–359. https://doi.org/10.1109/ICDE51399.2021.00037

[66] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering Regions of Different Functions in a City Using Human Mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Beijing, China) *(KDD '12)*. Association for Computing Machinery, New York, NY, USA, 186–194. https://doi.org/10.1145/2339530.2339561

[67] Nicholas Jing Yuan, Yu Zheng, and Xing Xie. 2012. *Segmentation of Urban Areas Using Road Networks*. Technical Report MSR-TR-2012-65.

[68] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 1655–1661.

[69] Mingyang Zhang, Tong Li, Yong Li, and Pan Hui. 2021. Multi-View Joint Graph Representation Learning for Urban Region Embedding. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (Yokohama, Yokohama, Japan) *(IJCAI'20)*. Article 611, 7 pages.

[70] Rui Zhang, Conrad Albrecht, Wei Zhang, Xiaodong Cui, Ulrich Finkler, David Kung, and Siyuan Lu. 2020. Map Generation from Large Scale Incomplete and Inaccurate Data Labels. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2514–2522. https://doi.org/10.1145/3394486.3403301

[71] Yipeng Zhang, Zhifeng Bao, Songsong Mo, Yuchen Li, and Yanghao Zhou. 2019. ITAA: An Intelligent Trajectory-Driven Outdoor Advertising Deployment Assistant. *Proc. VLDB Endow.* 12, 12 (aug 2019), 1790–1793. https://doi.org/10.14778/3352063.3352067

[72] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2020. Curb-GAN: Conditional Urban Traffic Estimation through Spatio-Temporal Generative Adversarial Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 842–852. https://doi.org/10.1145/3394486.3403127

[73] Yu Zheng, Tong Liu, Yilun Wang, Yanmin Zhu, Yanchi Liu, and Eric Chang. 2014. Diagnosing New York City's Noises with Ubiquitous Data. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Seattle, Washington) *(UbiComp '14)*. Association for Computing Machinery, New York, NY, USA, 715–725.

[74] Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie. 2011. Urban Computing with Taxicabs *(UbiComp '11)*. Association for Computing Machinery, New York, NY, USA, 89–98.

A ILLUSTRATIVE EVIDENCES

A.1 ACF_daily is a proper proxy for measuring predictability

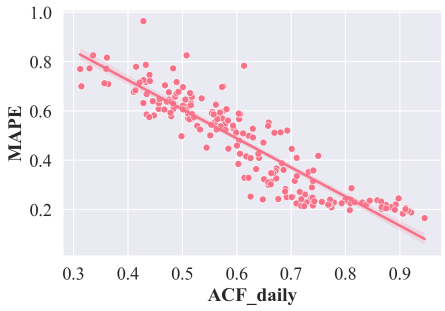


Figure 10: The ACF of different time series and the corresponding prediction error (Mean Absolute Percentage Error) given by STMGCN [18]. The prediction is conducted on the freight transport dataset (Sec. 5) for one-hour time slots. The ACF is computed at the daily scale, i.e., k is set to 24 in Eq. (1).

A.2 More data samples support more obvious daily regularity

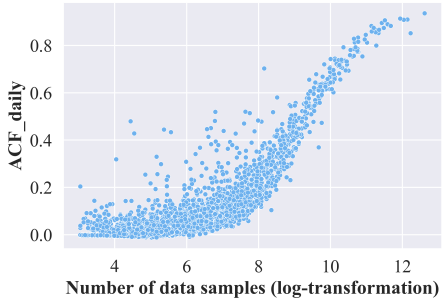
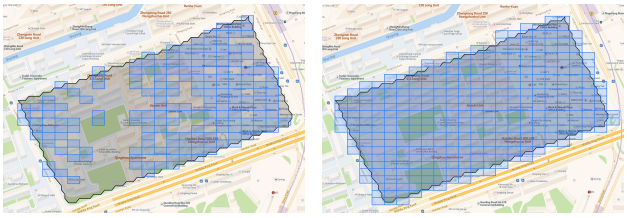


Figure 11: More data samples support more obvious daily regularity, i.e., ACF_daily. Every data point represents an atomic spatial element in the freight transport dataset.

A.3 Example of calculating specificity



(a) The serviced area (b) The total area

Figure 12: Examples of the serviced area and total area, the service specificity is calculated by 8-bit geohash.

B MULTI-OBJECTIVE NODE SWAPPING ALGORITHM

Algorithm 1 shows the predictability-specificity co-optimization algorithm.

Algorithm 1: Predictability-Specificity Co-optimization

Input: $D \in \mathbb{R}^{T\times N}$, $A \in \mathbb{R}^{N\times N}$, $vs, ts \in \mathbb{R}^{N\times 1}$, Max Epochs Eps, Geographic Constraints C, Predictability Weight w
Output: Pareto optimal solutions $\mathcal{Y}$

Initialize $bestACF \leftarrow 0$, $bestSpecificity \leftarrow 0$, $\mathcal{Y} \leftarrow \emptyset$;
Generate feasible solutions and append them to $\mathcal{Y}$;
while $Eps > 0$ **do**
  Sample a random number $p \in [0, 1]$ from uniform distribution ;
  if $p < w$ **then**
    Select the solution with the best predictability X from $\mathcal{Y}$;
  else
    Select the solution with the best specificity X from $\mathcal{Y}$;
  Get boundary set $\mathcal{B} \leftarrow$ `MovableBoundary(X,A,C)` ;
  for *each* $(u, v) \in \mathcal{B}$ **do**
    Move u to the cluster of v get new solution X' ;
    Calculate the gain of X' (i.e., *acf & specificity*) ;
    if *acf* > *bestACF* **or** *specificity* > *bestSpecificity* **then**
      update *bestACF* or *bestSpecificity* ;
      append new solution X' to $\mathcal{Y}$;
    $Eps \leftarrow Eps - 1$;
  Remove non-Pareto solutions in $\mathcal{Y}$;
return $\mathcal{Y}$

Function `MovableBoundary(X,A,C)`:
  Initialize $\mathcal{B} \leftarrow \emptyset$;
  for m *in* $\{1, 2, ..., M\}$ **do**
    Find nodes set U' that belongs to cluster m ;
    for *each node* $u \in U'$ **do**
      Find neighbour set V' of node u ;
      for *each node* $v \in V'$ **do**
        if u, v *belong to different clusters* **then**
          move u to the cluster of v and get X' ;
          if *constraints* C *are satisfying in* X' **then**
            append pair of nodes (u, v) to $\mathcal{B}$;
  return $\mathcal{B}$

Table 3: Dataset Statistics.

Datasets	Designated Driver	Freight Transport	Open Taxi
# Records	8,000,000	7,000,000	9,455,822
Time Span	2020.10-2021.08	2020.10-2021.08	2018.10-2019.05
Open Access?	Private	Private	Public
Road Data (OpenStreetMap)			
Type	8	8	N/A
# Roads	6867	6867	N/A
River Data (OpenStreetMap)			
# Rivers	1873	1873	N/A

C DATASET DESCRIPTION

Table 3 contains the statistics data of the following four datasets.

Designated Driver Dataset. The designated driver service dataset is sampled from a world-leading online transportation company, including the designated driver orders from Oct. 2020 to Aug. 2021 in one mega-city. The typical designated driver order takes place like this: after drinking alcoholic beverages, people are not allowed to drive and may seek help from the sober designated driver to take them home. We sample a certain percentage of the data and get a 10-month dataset with 8,000,000 records. Each record contains the start time and location (longitude and latitude).

Freight Transport Dataset. The freight transport dataset is also sampled from a world-leading online transportation company, including the freight transport orders in a metropolis from Oct. 2020 to Aug. 2021. The typical freight transport service is similar to online ride-sharing services. That is, users send orders online, and truck owners receive orders online and provide transportation services. We sample a certain percentage of the data and get the dataset with 7,000,000 historical records. Each record contains the start time and location (longitude and latitude).

Open Taxi Dataset. The taxi demand dataset is collected from Chicago's open data portal.[6] The dataset describes taxi trip records including the pickup time and location. Note that to protect privacy, the latitude and longitude of the trips are not recorded in the dataset; instead, the location is reported at the census tract level. Considering that census tracts already hold good spatial semantics, we use census tracts as the atomic spatial elements for clustering (without the need to do segmentation). We obtain the polygon boundaries of the census tracts from Chicago's open data portal[7].

Road and Obstacle Dataset. Road and obstacle data are collected from OpenStreetMap (OSM).[8] To produce fine-grained level road segmentation, we choose the majority of vehicle roads, primarily those classified as 'motorways', 'primary', 'secondary', and 'tertiary'. We extract river records from OSM waterway data.

D EXPERIMENTAL SETTING AND BASELINES

D.1 Baselines

For a fair comparison, *RegionGen* and all the baselines are tuned to generate the same number of regions (M). Specifically, M is set to 913, 724, and 95, respectively, for the designated driver, freight transport, and open taxi datasets. The baselines are as follows.

- ***Grid*** and ***Hexagon***: With poor spatial semantic meaning, we split the city into several grids/hexagons of equal size. The elements without spatiotemporal data will be filtered out.
- ***DBSCAN***: *DBSCAN* [10] is used for clustering transportation service orders' location points. It is a point clustering method and cannot output the shape of the generated regions directly.

Previous research has proposed several station-based clustering methods [4, 32], which aggregate nearby stations with similar spatiotemporal patterns. To adopt these methods, we take the atomic spatial elements as stations to generate regions by clustering.

- ***BSC*** [32]: The *Bipartite Station Clustering* (BSC) method groups stations into clusters based on their geographical locations (first partition) and transition patterns (second partition). As our datasets include demand records, we cluster the station by the demand matrix instead of the transition matrix in the second partition.
- ***GCSC*** [4]: The *Geographically-Constrained Station Clustering* (GCSC) method groups stations into clusters, making each cluster consist of neighboring stations with similar usage patterns.

D.2 Experimental Setting of Region Generation

The road and obstacle vectors in the entire city (about $80km \times 70km$) are converted into a binary raster with 8000×6000 pixels. We apply a small 5×5 dilation kennel as Yuan et al. [67]. In the graph generation component, atomic elements whose ACF_daily > 0.5 or area $> 5km^2$ are separate nodes. The geographic adjacent distance τ is 50m. The maximum area constraint is $5km^2$.

D.3 Setting of Spatiotemporal Prediction

To conduct demand prediction, we first select the spatiotemporal data in the last 10% duration in each dataset as test data and the 10% data before the test data as the validation test. All region generation methods are based on the data in the train set. We apply three state-of-the-art predictive models, including *STMGCN* [18], *STMeta* [55], and *GraphWaveNet* [61]). To capture spatial dependences, we introduce distance and correlation graphs as Wang et al. [55]. The distance graphs are calculated based on the Euclidean distance while the correlation graphs are computed by the Pearson coefficient of demand series. The hyperparameters of these deep models are fine-tuned by grid search and the hidden states of the STMGCN network are 64 (the dimension of spatiotemporal representations). The degree of graph Laplacian is set to 1. We use the Adam optimizer with learning rate = 1e-4.

D.4 Reproducibility

The designated driver dataset and the freight transport dataset are proprietary. The Chicago taxi dataset and the road and obstacle dataset are open access and thus the taxi demand results in Sec. 5.4 can be reproduced.

E FURTHER ANALYSIS

E.1 Initialization Methods

We here elaborate on three initialization methods that can generate feasible solutions.

D-Balance. Namely the *fast solver* in Sec. 3.3.2, we get the results by assigning the node weight with the amount of spatiotemporal data and minimize the edge-cut (to isolate the nodes with small degrees) while tolerating 5% unbalance.

Greedy. Graph growing is a widely adopted graph partition technique [3, 42]. The simplest growing method starts from a random node v, remaining nodes are assigned to the same cluster using a breadth-first search. The growth stops when a certain constraint transgresses. We extend the growing method by a greedy strategy to guide the node growth and optimize the objectives (e.g., predictability). First, we randomly select M nodes as the initial points

[6] https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew
[7] https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Census-Tracts-2010/5jrd-6zik
[8] We download OSM data from https://download.geofabrik.de/

of M clusters. Then we add unassigned nodes to the assigned clusters in turn by picking the unassigned node with the greatest gain. The following equation specifies the gain of appending node v:

$$gain(v) = \lambda \cdot \Delta\text{ACF} + (1-\lambda) \cdot \Delta\text{Specificity} \quad (11)$$

where ΔACF and ΔSpecificity are the change of the ACF_daily and specificity after appending node v into assigned sets. λ (usually set to 0.7 in practice) is a trade-off parameter between choosing better predictability solutions or better specificity solutions. Therefore, the *Greedy* solver converts the original problem into a single-objective optimization problem by the linear scalarization.

Fluid [47] is a community detection technique based on how fluids interact with one another and change size in their environment. We get the solutions by giving the 'aggregatable' graph $G = (V, E)$ and using a propagation-based approach with predefined cluster numbers.

E.2 Effect of Training Data Size

In Fig. 13, we use the previous 7 days, 14 days,... 140 days as training data to calculate the predictability and specificity of solutions, and evaluate them on testing data. We have the following observations: (i) The train ACF is usually higher than the test ACF, which indicates that the daily flow pattern will change over time, and thus the train ACF fluctuates when the training data size is small; (ii) More training data can bring better and more robust test ACF but calculating the ACF of long time series is time-consuming. We recommend 70 days as a good trade-off point to balance generalizability and computing efficiency.

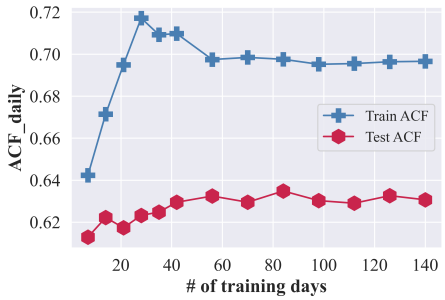


Figure 13: Effect of training data size

F FURTHER EVALUATION AND DISCUSSION

F.1 Evaluation on New Dataset

To evaluate the generalizability of *RegionGen*, we conduct further experiments on a new dataset, namely the taxi trip data in NYC [9]. We use the data from January 2013 to March 2013. Each record contains the pickup time and location (longitude and latitude). Table 4 lists the statistics of the Taxi NYC dataset.

We first conduct the region generation experiments with baselines. Then we compare the capability for the same downstream task (e.g., demand prediction) by utilizing *STMGCN*. We follow the

[9]https://data.cityofnewyork.us/Transportation/2013-Yellow-Taxi-Trip-Data/7rnv-m532

Table 4: Dataset Statistics of Taxi NYC.

Time Span	Records	Road Type	# Road	# Rivers
2013.01-2021.03	—-	—-	—-	—

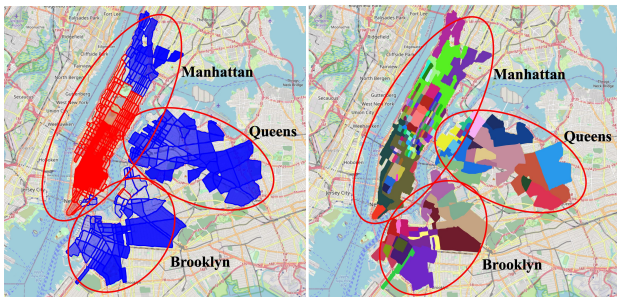
experimental setting as shown in Sec. D.2 and Sec. D.3. The results are in Table 5. For *RegionGen*, we report two solutions best in the ACF_daily and specificity metrics respectively. Table 5 shows that *RegionGen* achieves better ACF_daily and better specificity than the baselines and gets corresponding lower MAPE@Recall. The above experiments show that *RegionGen* still performs the best in the Taxi NYC dataset, demonstrating its good generalizability.

Table 5: Results on the Taxi NYC dataset. M is the number of regions. The best two results are highlighted (*best* is in bold and *italic*, second-best is in bold). The specificity of *DBSCAN* is not applicable (N/A) since it is a point clustering method.

	Taxi NYC (M=152)		
	ACF_daily	Specificity	MAPE@XX%
Baselines			
Grid	0.0000	0.0000	0.0000
Hexagon	0.0000	0.0000	0.0000
DBSCAN	0.5319	N/A	0.3303
BSC	0.6227	0.7499	0.2923
GCSC	0.6214	0.759	0.2979
RegionGen (best specificity)	***0.7242***	***0.8398***	***0.2727***
RegionGen (best ACF)	***0.7418***	***0.8276***	

F.2 Robustness Analysis on Different Subareas

To show the robustness of *RegionGen* for different subareas. We investigate different subareas of NYC, including Manhattan, Brooklyn, and Queens (we visualize these three boroughs in Fig. 14). We observe that Manhattan has more taxi demand than Brooklyn and Queens in Fig. 14(a). Therefore, we regard Manhattan as the hot subareas while Brooklyn and Queens as the relatively cold subareas. We conduct experiments on these two subareas respectively and the results list in Table 6.



(a) Heatmap (b) RegionGen

Figure 14: Heatmap and regions generated by *RegionGen*.

In Table 6, *RegionGen* achieves better ACF_daily and better specificity than the baselines and gets corresponding lower MAPE@Recall

in both hot and cold subareas. From Fig. 14(b), *RegionGen* generates many small regions in hot subareas (Manhattan), which is suitable for precise management. In relatively cold subareas (Brooklyn and Queens), *RegionGen* gives large regions to support better predictability. The above experiments show that *RegionGen* is robust to generate appropriate regions for different subareas.

Table 6: Results on the different subareas in NYC. M is the number of regions. The best two results are highlighted (*best* is in bold and *italic*, second-best is in bold). ACF is the abbreviation of ACF_daily. The specificity of *DBSCAN* is not applicable (N/A) since it is a point clustering method.

	Hot Subareas (M=118)			Cold Subareas (M=34)		
	ACF	Specificity	MAPE@XX%	ACF	Specificity	MAPE@XX%
Baselines						
Grid	0.3405	0.1315	0.0000	0.3405	0.1315	0.0000
Hexagon	0.3720	0.0000	0.0000	0.3720	0.2502	0.0000
DBSCAN	0.5834	N/A	0.0000	0.3533	N/A	0.0000
BSC	0.6602	0.8563	0.0000	0.4929	0.3807	0.0000
GCSC	0.6662	0.8708	0.0000	0.4659	0.3711	0.0000
RegionGen						
Best Specificity	***0.7864***	***0.9591***	***0.0000***	***0.5082***	***0.4255***	***0.0000***
Best ACF	***0.7875***	***0.9575***	***0.0000***	***0.5835***	***0.3771***	***0.0000***

F.3 Discussion on Predictability Objective

In *RegionGen*, we adopt the ACF_daily indicator for efficiently measuring the predictability of regions. The ACF_daily is computed by delaying the time series at the daily scale, which mainly captures the daily periodicity. However, in spatiotemporal transportation applications, weekly periodicity is also widely utilized [55, 68]. To explore whether maximizing ACF_daily can capture such weekly periodicity, we compute the ACF_weekly metric by delaying the time series to one week for every region in the freight transport dataset. We display the correlation between ACF_weekly and ACF_daily in Fig. 15. We observe that there is a strong positive correlation between these two metrics. Therefore, maximizing ACF_daily would also capture the weekly periodicity, and choosing ACF_daily as the fast proxy measurement for predictability is appropriate.

F.4 Applying to Dynamic Scenarios

Although *RegionGen* can finish region generation in two hours for a metropolis and satisfies the need for the 'T+1' update mode (Sec. 5.4), it may still face extreme dynamic scenarios (e.g., region hourly updates). An intuitive way to quick adaptation to dynamic scenarios is by reusing the previously generated solutions. That is, we could treat the previously generated solutions as the initial solutions rather than train from scratch (i.e., adopting the initialization methods introduced in Sec. E.1).

As shown in Fig 16, we first select a historical window to calculate the ACF_daily and get the corresponding solution X_t. We choose 70 days as the window size since it is a good trade-off point to balance generalizability and computing efficiency as shown in Fig. 13. Each time we slide the time series by d time steps, and we use the latest 70 days to compute ACF_daily and get solution X_{t+d}. Note that X_{t_d}

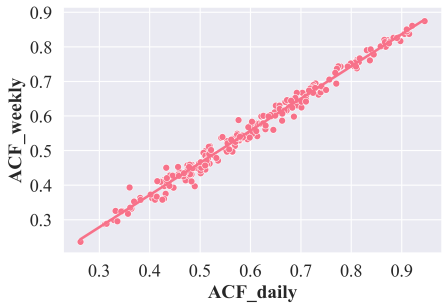


Figure 15: The correlation between ACF_daily and ACF_weekly. The analysis is conducted on the freight transport dataset (Sec. 5) for one-hour time slots. The ACF_daily and ACF_weekly are computed at the daily and weekly scale respectively (i.e., k is set to 24 and 168 in Eq. (1).

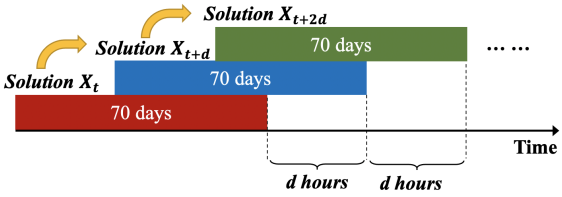


Figure 16: An example of dynamic scenarios.

iteratively improves from X_t. We conduct this experiment in the Taxi NYC dataset. Fig. 17 shows the results.

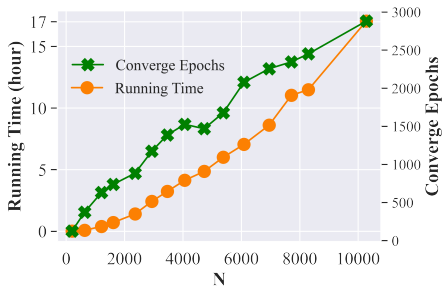


Figure 17: Refinement speed vs. d

In Fig. 17, it is evident that the refinement time increases gradually with the increase of d. This can be attributed to the fact that the time series exhibit greater changes with larger sliding steps. Notably, when d is 1, the refinement process requires only a minimal time of XXX minutes. This implies that *RegionGen* can promptly generate the most recent regions within XXX minutes of receiving the latest spatiotemporal data. The rapidity of the refinement process is primarily due to the minor changes in the time series of dynamic scenarios. Consequently, *RegionGen* is suitable for dynamic scenarios by leveraging previous results and partially refining them.

F.5 Limitations and Future Work

At present, the proposed obstacle-aware road map segmentation method takes the road map and obstacle data from OpenStreetMap.

However, In cases where the data quality from OpenStreetMap deteriorates in rural or remote areas, the segmentation results may not be suitable. In such circumstances, the utilization of the high-quality maps generated based on external data sources (e.g., high-resolution aerial images [70] or GPS Traces [40]) may benefit *RegionGen*. Besides, external spatial information (e.g., Point of Interest) may also benefit the process of region generation. As shown in Fig. 1, Region A, B, and C all contain university campuses, it is natural to aggregate them. Though *RegionGen* also aggregates these three regions since the merged region is more predictable, it still lacks the utilization of POI.

In future work, we will extend the ability of our framework to handle incomplete road maps or obstacles data and try to incorporate more spatial information (e.g., POI) for better spatial semantics.