Dynamic Spatio-temporal Access Queries using Semi-Supervised Regression

Chris Conlan
Department of Computer Science
University of Warwick
Coventry, UK
Chris.Conlan@warwick.ac.uk

Teddy Cunningham
Department of Computer Science
University of Warwick
Coventry, UK
Teddy.Cunningham@warwick.ac.uk

Hakan Ferhatosmanoglu*
Department of Computer Science
University of Warwick
Coventry, UK
Hakan.F@warwick.ac.uk

***Abstract*—Understanding the cost of accessing services in a transit network, and how this varies spatially and temporally is vital for transport agencies to make effective decisions. However, to understand this at the city-scale typically demands the computation of a very large number of shortest path queries, which is computationally infeasible in a practical setting. In this work we define the notion of an access query, an analytical query which returns the aggregate access costs to a set of points of interest within a given time interval. To solve the computational bottleneck, we develop a solution that uses semi-supervised machine learning to efficiently compute these aggregate access costs using a gravity-model. The solution dynamically generates a descriptive representation of the connectivity between origins and destinations in a multi-modal network, and dynamically labels a small subset of the overall trips which are used to form a target vector for the learning algorithm. We also consider the fair distribution of access across spatio-temporal dimensions. The solution can reduce processing times by up to 97%, while maintaining high levels of accuracy; the predicted journey times to services are accurate to within 3.3 minutes, and a high level of correlation (85%) to the ground truth is achieved.**

***Index Terms*—Urban Analytics, Accessibility, Semi-Supervised Learning, Spatiotemporal Data**

I. Introduction

Understanding the accessibility provided by a city's transit system to key services, such as hospitals or schools, and understanding how it varies spatially, temporally, and across demographics is fundamental to ensuring that public bodies provide an effective, fair, and equitable society [1]. A policy maker may ask analytical queries (henceforth, access queries - AQs) such as:

1) What is the average travel time to an important service (e.g., hospital), and how does this vary spatially and temporally?
2) Considering the monetary cost and the inconvenience of transit (e.g., changing buses), what is the overall accessibility to key services and how does this vary spatially and temporally?
3) Which geographic areas are most at risk? For example, are there unemployed people unable to reach job centers? Does the varying transit schedule in some places restrict or prevent access at particular times of the day?
4) Are the accessibility benefits provided by the transit system fairly distributed between, and within, key demographic groups?

Computing a temporal origin-destination access matrix (TODAM) can provide the underlying aggregate data for all of these types of access queries. The TODAM aims to capture the expected access cost between each of a city's geographic locations (often given by its census tracts) and their respective points of interest (POIs) that people want to travel to, over a discrete time scale. It is inherently a large and unwieldy matrix that is expensive to compute and maintain over time. Each entry in the TODAM is a trip with an origin, destination, and journey start time. To populate the TODAM, a measure of each trip's accessibility, its access cost, is calculated and recorded. Naïvely, and in practice [2], [3], this can be performed by running a shortest path query (SPQ) in a multi-modal transit network for each trip. This can result in many millions of SPQs at the city scale given the increasingly fine granularity of space and time. While a single SPQ can be efficient in isolation, performing millions of these queries becomes extremely costly. We have experienced these performance challenges while we supported Transport for West Midlands (TfWM) in locating the initial COVID-19 vaccination sites in December 2020. Our focus was on understanding accessibility for the most clinically vulnerable, and ensuring it was adequate and fairly distributed. Computing a TODAM for POIs in the West Midlands (approximately three million people) for a given time interval (e.g., weekday AM peak, PM peak) took up to 36 hours in a parallel CPU infrastructure. Measuring all-pairs accessibility at different times throughout the day in another study resulted approximately four billion trips. Using traditional GIS tools to compute the TODAM would take over 100 days [4]. This is clearly a major computational bottleneck for answering AQs efficiently.

In GIS and transportation domains, accessibility analysis is typically performed as statistical analysis on static datasets [5]–[9]. However, practitioners, such as transport policy mak-

We thank Transport for the West Midlands and our collaborators there, particularly Charmaine Grant. We thank Artemy Bulavin for his contributions to some of the underlying implementation. This research is supported in part by the UK Engineering and Physical Sciences Research Council under Grant No. EP/L016400/1.

*Hakan Ferhatosmanoglu is also with Amazon Web Services. This work was performed at the University of Warwick and is not associated with Amazon.

ers, need to operate in a dynamic environment and test new policy scenarios, such as optimally locating a new school to improve accessibility for families or introducing new bus stops to avoid "access deserts". In order to support dynamic AQs, the TODAM needs to be recomputed with every spatio-temporal change to the system. There is a need for a scalable and dynamic solution that provides timely information and fast responses to AQs. Improving the efficiency of a single SPQ is not enough to enable truly dynamic querying. Dynamic accessibility analysis also needs to reduce the number of SPQs that need to be solved when calculating the TODAM. Only then can the operation become sufficiently efficient to enable dynamic AQs, and be able to scale to large cities or regions.

To address these challenges, we develop a semi-supervised regression (SSR) based solution that significantly reduces the SPQ workload needed to compute accessibility measurements at the city-scale. A small subset of locations are selected for labeling. Labeling involves drawing a set of representative trips for each of these locations (e.g., origin points) from the TODAM and calculating the access costs using SPQs. These costs are aggregated to the origin-level providing an 'access score', which forms the target vector for the learning algorithms. The access scores are then inferred by the model for the remaining, unlabeled, locations. The solution embeds the widely used gravity measurement of accessibility into the construction of the TODAM to significantly reduce redundant downstream computations, and then again into the aggregation of the TODAM when forming the feature and target vectors. These aggregations translate the data to the origin-level providing efficiency boosts during the learning phase, and with negligible information loss.

One of the major challenges is efficient feature generation, particularly given that the underlying spatio-temporal data that describes accessibility is large and costly to query. To address this, our solution dynamically forms a feature vector for each OD pair in the TODAM, which is then aggregated to the origin level. The underlying network, scheduling, and location data are pre-processed to form efficient data structures that can reveal highly descriptive information about two locations' respective connectivity. These structures are called transit-hop trees and, when retrieved for a particular location, they instantly provide the set of other locations in the city that are accessible using a short foot journey and/or single transit journey, as well as data about the connectivity (e.g., route frequency, travel time). When these structures are centered on both the origin and destination points, they provide an effective way to efficiently generate features describing connectivity.

We evaluate our solution on real-word data sets for two cities in the UK, measuring access to schools, hospitals, COVID-19 vaccination centers, and job centers. We test several SSR algorithms, from classical methods (e.g., COREG) to deep-learning methods (e.g., graph neural networks). Our solution makes significant reductions in the processing time (up to 97%) while still maintaining accurate accessibility measurement inferences. For instance, at a 3% labeling budget, travel times to schools are accurate to within 3.3 minutes. Our methods are effective on the more complex generalized access cost (which combines multiple relevant costs), where we report correlations between ground truth and predicted values of 85%.

The contributions of this work are summarized as follows:

- Formalization of dynamic AQs in a data management context using gravity-based measurements.
- Proposal of a machine learning (ML) based approach as an alternative to explicit management and querying of spatio-temporal data for accessibility analysis.
- Demonstration of the efficacy of the SSR-based solution for efficiently computing accessibility measurements at the city scale.
- Definition of novel data types to efficiently compute connectivity between two places, and a method to pre-compute these data types from network, scheduling, and location data.
- Introducing the notion of a fairness index to determine how fairly access is distributed in a region, and show how our method accurately predicts this measure.

II. Related Work

Over several decades, researchers in GIS and transportation domains have been measuring accessibility using a variety of methods – see survey papers [10]–[12].

Temporal Accessibility Studies. Recently, the temporal dimension has become an essential part of accessibility studies [2], [3], [13], which has further increased the computational impact of accessibility analysis. A comprehensive summary is provided by [14]. It has been shown that accessibility patterns measured at a fine temporal scale can vary significantly, even over a short time window [3]. This can reveal important patterns in accessibility, such as 'food deserts' emerging as people cannot access supermarkets at certain times [2]. While the notion of a travel-time cube (similar to the TODAM) enables a range of spatio-temporal analyses, computational challenges remain [15]. Recent approaches incorporate more varied sources of data on observed transit times to more accurately capture spatio-temporal patterns [16], [17].

Computing Accessibility. Prior research limits the geographic scope of their analysis to ensure it is computationally feasible [18], whereas our solution can scale to large networks. To improve efficiency, parallel processing has been used to speed up accessibility calculations [13], [19]. While this may be a viable strategy for some static data analysis, it is still not efficient, e.g., an all-pairs matrix is reported to take four hours to complete using 64 CPUs. Also, parallelization can benefit an SSR approach too, as the majority of the runtime is in labeling. An adapted all-pairs shortest path algorithms has been developed to compute citywide all-pairs accessibility [20], taking six days to run for a large transit network. We aim to show how targeted queries can be processed much more efficiently.

Machine Learning for OD Travel Time Estimation. A related problem is real-time OD travel time estimation. Recently several papers have applied deep learning to the task and shown significant results [21]–[24]. The challenge has been to discover meaningful feature representations that can be computed efficiently to enable the machine learning models to learn accurate mappings. It is shown that the pre-processing of spatio-temporal road network data and novel representations of historically observed traffic patterns are an effective approach [21]. To our knowledge, no work has applied machine learning to the computation and management of accessibility analysis.

III. Defining Dynamic Access Queries

In this section, we define the data specifications and introduce a gravity model for measuring accessibility which is used to define the structure of the TODAM. We then describe how to generate an access score from the fully populated TODAM, which provide answers to dynamic AQs.

A. Preliminaries

The overall region is discretized into a set of zones, $z_i \in \mathcal{Z}$, which can be defined by public information such as census tracts. Each $z_i \in \mathcal{Z}$ is represented by the latitude/longitude of its geographic centroid. To denote a POI we use p_i, and $\mathcal{P}$ is the full set of POIs. Each $p_i \in \mathcal{P}$ is attributed with, as a minimum, its latitude/longitude coordinates and a category (e.g., school, hospital, job center).

The road network is modeled as a graph, $\mathcal{G}(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. A transit network (e.g., bus network) augments $\mathcal{G}(\mathcal{N}, \mathcal{E})$ with timetable information to provide the temporally varying transit times in the network. This information is often publicly available (e.g., General Transit Feed Specification, GTFS), and contains information about stops, routes, and individual departure and arrival times. We refer to this data as $\mathcal{F}$.

We divide the time domain into a series of time intervals $\mathcal{V}$. Each $v \in \mathcal{V}$ is given as $v = [t_s, t_e, t_d]$, where t_s and t_e are the start and end times, and t_d is the day of the time interval. These time intervals are labeled to denote the popular times for which it is important to assess accessibility to key services. For example, [7am, 9am, Tuesday] represents the typical rush-hour peak of a weekday morning.

B. Measuring Accessibility

We employ a location-based measure of accessibility that works at the aggregate level (e.g., zone level). This involves analyzing aspects of land use and travel impedance (i.e., how difficult it is to get from one place to another). Due to lower data requirements and ease of interpretation, an aggregate measure is often preferred to person-based approaches [10]. Specifically, we employ a gravity model, which are widely adopted in literature [4], [5], [7], [25], [26]. Gravity models neatly capture the interplay of both land-use patterns and transit system coverage, and they are versatile in measuring aspects of social exclusion [12] and appropriate for larger networks [4]. The original equation [27] is still central to gravity models, where accessibility to a particular POI, p_j, from a zone z_i is a function of its impedance in $\mathcal{G}$ weighted by its attractiveness (to the residents at z_i), known as the access cost. An 'accessibility score' for each z_i is normally calculated by aggregating the access cost over all $p_j \in \mathcal{P}$.

C. Temporal Origin-Destination Access Matrix

In practical terms, a set of trips is generated from each zone, z_i, to each POI, p_j at varying start times t. Each trip's access cost is measured, and to calculate accessibility these costs are aggregated at the zone-level using some form of the Hansen equation. The TODAM is a three-dimensional matrix that captures all of these trips. We formalize a method to integrate the gravity model into the construction of the TODAM, which uses an attractiveness score between z_i and p_j, as defined below, to restrict the number of trips sampled. In effect, this moves the Hansen equation downstream. In doing so, when the access costs are calculated and aggregated, the gravity model has been computed, significantly reducing the size of TODAM.

The full TODAM is defined as M_f. It is a matrix of dimensions $|Z| \times |P| \times |R|$, where R is a set of randomly generated start times drawn from v, determined by a per hour sample rate. When TODAM is constructed using the gravity model it is defined as M_g. A binary matrix, M_b, of the same dimensions as M_f is defined, and M_g is formed by selecting from M_f where the corresponding entry in $M_b = 1$.

To populate the M_b matrix, we must define attractiveness, which denotes how likely someone in a zone z_i is to visit a POI p_j. This score determines how many trips are sampled from R for a (z_i, p_j) pair. The attractiveness score can be given by domain knowledge, learned from real data, or calculated on-the-fly (e.g., by using a distance decay function [12]). The score is then normalized over all $\mathcal{P}$ for each $z_i \in \mathcal{Z}$. Thus, each (z_i, p_j) has a score between 0 and 1, given as α_{ij}, which denotes p_j's relative attractiveness to the residents in z_i.

M_b is populated as follows: When $\alpha_{ij} = 0$ then no trips between z_i and p_j are generated, thus $M_b^{i,j,:} = 0$. When $\alpha_{ij} > 0$, then some trips between z_i and p_j are sampled from R, denoted by $r^{i,j}$, thus $M_b^{i,j,r^{i,j}} = 1$. $r^{i,j}$ is proportional to α_{ij} and is governed by a probability function. We describe our approach in Section V.

To naïvely populate either M_f or M_g, each trip has its access cost calculated. This is typically performed by running a SPQ in $\mathcal{G}$. We calculate two access costs: journey time (JT) and generalized access cost (GAC). JT is defined formally as: $c(o, d, t) = AT(d) - t$, where AT is a function that returns the arrival time at a given location. GAC is a popular approach [5], [6], [8], [18], [28] that calculates access as a combination of monetary, inconvenience, and time costs. We use the definition given by the UK Department for Transport [29], formally[1]:

$$c(o, d, t) = \lambda_1 TAN + \lambda_2 WT + \lambda_3 IVT + \lambda_4 ET + TP + \frac{FARE}{VOT} \quad (1)$$

where λ_i is a non-negative weighting factor.

[1] TAN: time to access G; WT: waiting time; IVT: in-vehicle time; ET: egress time; TP: transfer penalties; FARE: fare; VOT: value of time conversion factor

D. Accessibility Measures

We formalize a set of measures which, when calculated on M_g, implements the gravity model of accessibility. These measures are calculated at the zone-level, and are typically mapped to provide a visual analysis.

Mean Access Cost (MAC). The expected level of accessibility, according to the gravity model, of a zone, z_i:

$$MAC_i = \frac{\sum_j \sum_k (c(z_i, p_j, r^{i,j,k}))}{|M_g^{i,:,:}|} \tag{2}$$

Access Cost Standard Deviation (ACSD). The standard deviation of all observed access costs at the zone-level is given to represent the variation in accessibility in the temporal domain.

Accessibility Classification (AC). In isolation, aggregate measures can be hard to interpret, particularly for GAC [8]. Hence, we classify a zone's accessibility, as in [2]. The classification rules are: low MAC and low ACSD receives a class "best"; high MAC and low ACSD receives a class "worst"; low MAC and high ACSD receives a class "mostly good"; high MAC and high ACSD receives a class "mostly bad". Low means below average, high means above average.

Fairness Index. We introduce the concept of the fairness index, which seeks to determine how fairly MAC is distributed across spatio-temporal dimensions and to understand if particular zones have significantly worse provision of access than others. The fairness index can be further weighted by zone-level demographic data to understand if particular groups are being unfairly served. In this paper we adopt the Jain's index (developed for computer networks [30]) over MAC as a fairness index.

IV. Dynamic Access Queries using Semi-Supervised Regression

Fully populating the TODAM requires an SPQ in $\mathcal{G}$ for each entry, each costing $O(|\mathcal{E}| \times \log |\mathcal{N}|)$. As the TODAM is typically very large, this is a major computational bottleneck in efficiently answering dynamic AQs. The average run time of each SPQ on our real-world network data was 0.018±0.016s [2]. To illustrate the bottleneck, consider the following example. In a city of 3000 census tracts, with 100 POIs, with each OD pair generating five different trip start times, the total runtime would be 7.5 hours on standard CPU architecture (without parallelization). In practice, many AQs may be required to measure accessibility at each time interval. We aim to compute the access measures in Section III-D using only a fraction of the number of SPQs.

Our solution, which uses SSR methods and includes dynamic feature generation and data labeling, has four constituent parts, which are summarized in Fig. 1. In the following sections, we describe each component in more detail and discuss the technical challenges.

A. Feature Extraction: Offline Processing

The main challenge in offline feature extraction is to transform the large transit and spatial data into efficient structures to describe a (z_i, p_j) relationship, in which p_j is associated to its zone z_j. At its heart is a notion we call the "transit hop". An outbound transit hop is any viable (o, d, t) journey through $\mathcal{G}$ composed of a short foot-journey from o and/or a transit-journey of any length to d. An inbound transit hop reverses the foot and transit journey components. Transit hops are pre-computed at the zone level and saved to a data structure called a transit hop-tree - we depict this structure in Fig. 2B. At its root is a zone, z_i, and associated leaves are all the zones reachable after one hop, with data held on each leaf to describe the connectivity in $\mathcal{G}$ (route frequency and average journey time). These structures are versatile, and enable important information to be accessed efficiently. They can also be chained easily to provide information after multiple (h) hops. Fig. 2A maps a (z_i, z_j) query along an outbound transit-hop tree centered on z_i and inbound transit-hop tree centered on z_j. We can observe how much information about the potential connectivity between these two points is instantly available. As connectivity data on each of the mapped points is available it enables a wide range of features to be calculated.

Transit-Hop Tree Generation. We now describe the procedure to form transit-hop trees for a zone, z_i for a time interval v_i. $OB_{z_i}^{v_i}$ denotes the transit-hop tree for outbound hops, and $IB_{z_i}^{v_i}$ the tree for inbound hops.

An isochrone (see Fig. 2C) for each $z_i \in \mathcal{Z}$ is pre-computed. The isochrone is calculated in $\mathcal{G}$, given an acceptable walkable time in seconds (τ) and a walking speed (ω). This outputs a set of shapefiles representing the walkable area around each z_i. We denote the full set of walking isochrones as $\mathcal{W}$, where W_i is the isochrone for z_i. F_{stops} is intersected with W_i to retrieve the set of bus stops that are walkable from z_i. For each bus stop, all the services that pass through it during v_i is retrieved from F_{trips}. Then, for each service, each subsequent bus stop is 'visited' (use preceding stops for inbound). The zone, z_j, for each of these bus stops is added as a leaf to $OB_{z_i}^{v_i}$. If it already exists, the connectivity data is updated to provide details of this newly discovered route. The journey times for each discovered route are recorded in a list on the leaf and a counter on the leaf is incremented by one. When each bus stop has been processed the procedure ends, and the tree is saved such that it can be retrieved efficiently.

B. Feature Extraction: Online Computations

We describe how a feature vector is efficiently computed. Given a (z_i, z_j) pair, $OB_{z_i}^{v_i}$ and $IB_{z_j}^{v_i}$ are retrieved and interchanges are identified. Interchanges are practical to compute online, and reveal how possible routes might be constructed between (z_i, z_j) providing even more information on potential connectivity.

1) Interchanges Identification: An interchange occurs when any $z_k \in OB_{z_i}^{v_i}$ is within walking distance of any $z_k \in IB_{z_j}^{v_i}$, allowing a passenger to connect to that service. These are

[2] These measurements were taken on the dataset described in Section V averaged over all POI types.

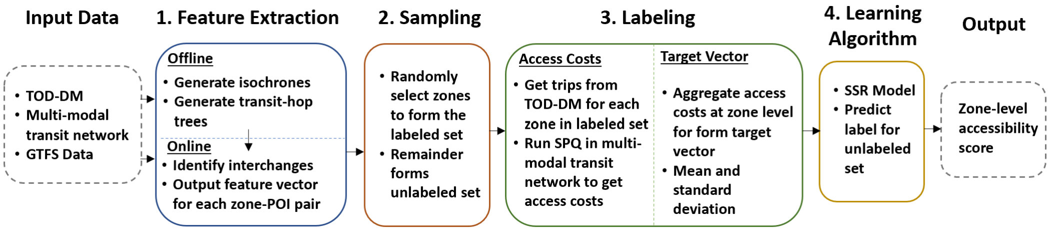
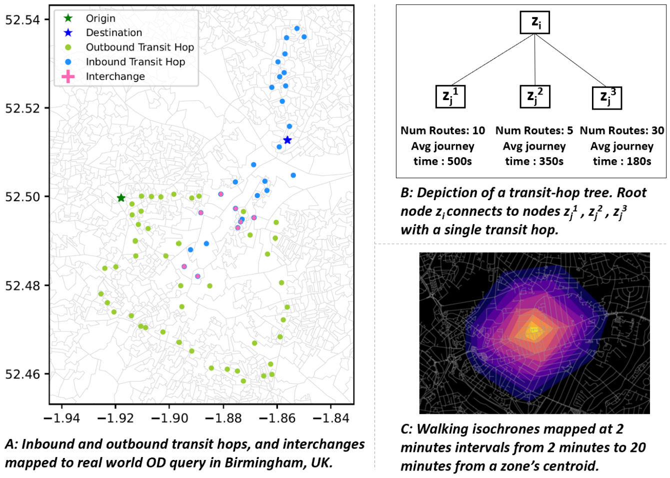

Input Data
1. Feature Extraction
2. Sampling
3. Labeling
4. Learning Algorithm
Output
• TOD-DM
• Multi-modal transit network
• GTFS Data
Offline
• Generate isochrones
• Generate transit-hop trees
Online
• Identify interchanges
• Output feature vector for each zone-POI pair
• Randomly select zones to form the labeled set
• Remainder forms unlabeled set
Access Costs
• Get trips from TOD-DM for each zone in labeled set
• Run SPQ in multi-modal transit network to get access costs
Target Vector
• Aggregate access costs at zone level for form target vector
• Mean and standard deviation
• SSR Model
• Predict label for unlabeled set
Zone-level accessibility score

Fig. 1: Diagram of SSR learning procedure to generate accessibility metrics.



A: Inbound and outbound transit hops, and interchanges mapped to real world OD query in Birmingham, UK.

B: Depiction of a transit-hop tree. Root node z_i connects to nodes z_j^1, z_j^2, z_j^3 with a single transit hop.

C: Walking isochrones mapped at 2 minutes intervals from 2 minutes to 20 minutes from a zone's centroid.

Fig. 2: Examples of transit-hops, transit-hop trees and walking isochrones.

shown in Fig. 2A. To compute this, a k-NN ($k = 1$) search is made for each $z_k \in OB^{v_i}_{z_i}$ on $IB^{v_i}_{z_j}$ to retrieve the nearest-node pairs. For each of these pairs, the walking isochrone for one is retrieved to test if the other intersects. When they intersect, it is recorded as an interchange.

2) Feature Vector Definition: To describe the (z_i, z_j) relationship: We instantiate a binary field that determines if d is reachable from o within h transit-hops. The closest point to d in $OB^{v_i}_{z_i}$, and closest point to O in $IB^{v_i}_{z_j}$ can be calculated easily and represented as their Euclidean distance and connectivity features (average travel time and number of transit routes). From the list of interchanges, we calculate which points are closest to o and d. These points can also be represented by their Euclidean distance and connectivity features. We can consider the leaves in $OB^{v_i}_{z_i}$ and $IB^{v_i}_{z_j}$, and identify those with the highest number of connections (e.g., high frequency routes). Then calculate how close we can get to d (or o) by traveling on these high-frequency routes, and count how many high frequency routes have interchanges. The percentage of reachable nodes in $\mathcal{G}$ after h transit hops can be calculated easily.

C. Data Sampling

The data in M_g are split into two sets for SSR training: labeled ($\mathcal{L}$) and unlabeled ($\mathcal{U}$). The size of $\mathcal{L}$ is determined by the sampling budget, β, a value between 0 and 1.

Data is aggregated to the zone level (e.g., origin level). This is convenient as the access measures are also calculated at this level. These measures are used directly as the target vector. This approach has the benefit of being efficient as $|\mathcal{L} \cup \mathcal{U}|$ will be small. The (z_i, p_j) level (e.g., OD level) may be considered as this retains more granular information on the feature vector, but the target vector must be aggregated to the zone-level post training. This is difficult as performing a weighted aggregation of standard deviations is computationally challenging and accuracy is hard to ensure. Learning at the trip-level (e.g., (o, d, t) level) is not viable, as calculating a feature vector for each trip would not be any more efficient than calculating the access cost for each trip.

To form the sets $\mathcal{L}$ and $\mathcal{U}$, we perform random sampling on $\mathcal{Z}$ based on β, which we assume gives a reasonable level of geographic coverage across the area of study. Active learning strategies may be explored to ensure coverage and to capture aspects of uncertainty. We note that the feature vector is generated on the OD level. For training, it is aggregated to the origin-level using a mean function weighted by α_{ij}, which applies the same weighting factor as the gravity-based access measures.

D. Semi-Supervised Regression and Inference

For labeling, each zone is selected in $\mathcal{L}$ and all of its respective trips are retrieved from M_g. For each region, an SPQ is run in $\mathcal{G}$ to calculate its access cost. Open Trip Planner[3] is used to dynamically run these queries, which provides realistic routes to (o, d, t) queries. It has been widely used to calculate access costs [**?**], [3]. These access costs are then aggregated back to the zone-level using the mean and standard deviation, which forms the target vector.

An SSR model is trained on $\mathcal{L}$ and used to infer the labels for $\mathcal{U}$ given their feature vectors. There are a wide range of available methods for this, many of which are discussed in Section V. The fully labeled $\mathcal{L} \cup \mathcal{U}$ allows us to efficiently calculate all of the accessibility measures and thus efficiently answer dynamic AQs.

E. Efficiency of the Solution

We analyze the efficiency of each step of the overall framework that is performed online.

The feature set is calculated online for each instance (z_i, p_j). Interchanges are calculated with a k-NN query that fits to the leaves of $OB^{v_i}_{z_i}$ and is applied to the leafs of $IB^{v_i}_{z_j}$.

[3] https://download.geofabrik.de/europe/great-britain/england/west-midlands.html

In the worst case, either of these structures could have $|\mathcal{Z}|$ leaves (although this is far fewer in reality). The cost reduces to $\mathcal{O}(|\mathcal{Z}| \times \log(|\mathcal{Z}|))$. Each leaf in $OB^{v_i}_{z_i}$ and $IB^{v_i}_{z_j}$ is accessed to calculate features, such as the distances to o and d. The cost for this operation reduces to $\mathcal{O}(h \times |Z|)$. In practice h is very low (e.g., 1 or 2). The total cost for to compute the feature vector for each (z_i, p_j) is $\mathcal{O}((|\mathcal{Z}| \times \log(|\mathcal{Z}|)) + (h \times |\mathcal{Z}|))$.

For labeling, a Dijkstra-like algorithm is typically used to provide the access costs, giving an overall labeling time complexity of $O((\beta|M_f|) \times |E| \log |\mathcal{N}|)$. This labeling cost is typically the largest component in the overall solution, and it is proportional to β. When β is low, the overall cost for measuring accessibility is low, and thus scalability is achieved for large networks.

V. Experiments

We run experiments on two UK city data sets, Birmingham (population: 1.14 million) and Coventry (population: 650,000). The experiments are all run on a GPU server (4 GPU cores, 500GB) and implemented using Python 3.7. We use PyTorch as our deep learning framework.

A. Setting

Data. The region of study is determined by shapefiles divided into census tracts[4], which forms the set of zones $\mathcal{Z}$. Coventry has 1014 zones, Birmingham has 3217. We use the GTFS feed provided publicly by TfWM[5].We use four different POI sets: schools, hospitals, COVID-19 vaccination centers, and job centers. The locations are scraped from the web. The number of POIs is shown in Table I. The results are reported for a single time interval - weekday AM peak. Scheduling data is selected between 7am to 9am on Tuesday May 10th 2022. A negative exponential distance decay function [10] is used to determine $r^{i,j}$. We use $\tau = 600$ and $\omega = 4.5kph$.

Models. We implement the following models: OLS regression (OLS), mean teacher (MT) [31], COREG [32], multi-layer perceptron (MLP), and graph neural network (GNN). We use some of the code provided by [33] for MT and COREG. In all implementations, a feature set is given for all $\mathcal{L} \cup U$, and the target vector is given for $\mathcal{L}$. The goal is to learn the labeling for U. We test the following values of β: 30%, 20%, 10%, 7%, 5% and 3%. For GNN, the adjacency matrix is calculated using the Euclidean distance between each $z_i \in \mathcal{Z}$, and then normalized using the Gaussian thresholded approach.

Performance Measures. We calculate the mean absolute error (MAE) and Pearson correlation coefficient (corr) of MAC and ACSD between the ground truth and predicted labels. In the case of AC, we calculate the accuracy, which is the percentage of correctly predicted classes. The difference between the ground truth and the predicted fairness index is the Fairness Index Error (FIE).

B. Experimental Results

TABLE I: Table showing size of different matrix variants for both cities and all POI types.

	Birmingham				Coventry			
	$\lvert\mathcal{P}\rvert$	Full	Gravity	% Red.	$\lvert\mathcal{P}\rvert$	Full	Gravity	% Red.
School	874	169,014,120	3,576,653	97.9	230	13,993,200	795,375	94.3
Hospital	56	10,829,280	2,314,256	78.6	6	365,040	142,864	60.9
Vax Center	82	15,857,160	2,133,585	86.5	22	1,338,480	322,188	75.9
Job Center	20	3,867,600	972,693	74.9	2	121,680	121,664	0.0

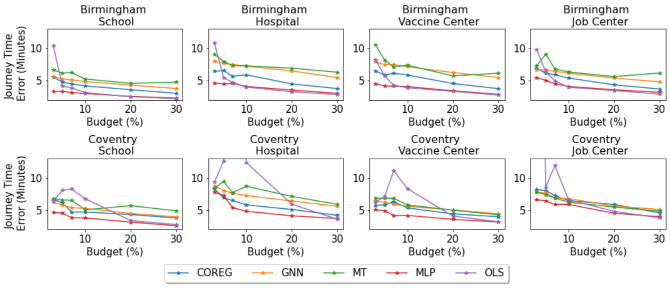


Fig. 3: JT errors for SSR solution across all POI types

1) Matrix Composition: Table I compares the size of the full matrix to the gravity-constructed matrix to observe the reductions achievable. In the most extreme case (School, Birmingham), the full matrix consists of nearly 170m trips. Assuming a per trip runtime of 0.018s to calculate the access cost, it would take 850 hours to process. Applying the gravity model reduces the size to 3.57 million, a 97.9% reduction.

In Birmingham, the reduction achieved from using the gravity model across the different POI sets is 84.5% with an average POI set size of 258, compared with an average reduction of 57.8% in Coventry (on an average of 65 POIs). Embedding the gravity model into the construction of the TO-DAM can make savings, and can be applied when performing any accessibility analysis, not just using SSR model.

2) SSR Solution Performance: Fig. 3 presents the mean error reported on journey times. MLP tends to be the best performer. As expected, there is some deterioration of the error term as the budget gets smaller. However, in most cases, this is not a severe drop off. In general, Birmingham can tolerate lower budgets than Coventry. With a 3% labeling budget in Birmingham, the reported errors are relatively low, while in Coventry a 7% labeling budget keeps errors relatively low.

OLS reports similar errors to MLP at a higher labeling budgets. However, at smaller budgets, it is inconsistent, and produces occasionally much worse results. This shows that deep learning approaches can find meaningful patterns in smaller datasets. The more bespoke SSR methods, such as COREG and mean teacher are not particularly competitive compared to MLP. GNN also fails to be competitive.

The GAC is generally more meaningful to policy makers as it also accounts for monetary cost and inconvenience. Fig. 4 shows the performance metrics for GAC for vaccination centers (the results are quite consistent across all POI types). Firstly, we consider the MAC correlation. Once again, MLP tends to outperform all other models. In Birmingham, the correlation is generally high, even at lower labeling budgets.

[4] https://geoportal.statistics.gov.uk/datasets/output-areas-december-2011-boundaries-ew-bgc

[5] https://api-portal.tfwm.org.uk/docs

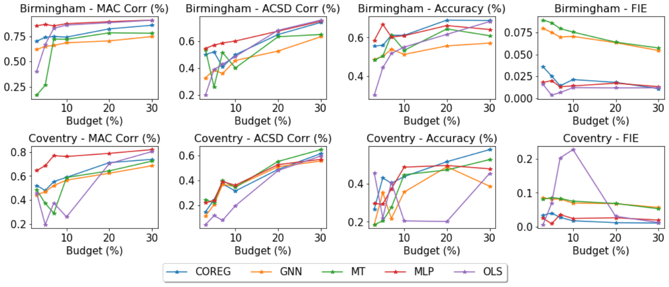


Fig. 4: Reported performance of SSR solution across different models on each location on vaccination center POIs.

In Coventry, the correlation drops off more noticeably after 7%, although it still remains strongly positive. These strong correlations demonstrate the overall patterns of accessibility are being captured accurately.

The ACSD correlation is less reliable, and we observe a more significant drop off at lower budgets across all models. In Birmingham, MLP tends to perform the best, while all of the models report similar performance in Coventry, except for OLS. While some drop-off exists at lower budgets for MLP in Birmingham, it remains strongly positive for all reported budgets. In Coventry, while all models report a positive correlation, it tends to be a weaker correlation, even for the best performing models.

The contrasting results between Coventry and Birmingham at lower budgets can be explained by the impact of walking only trips. When a zone is associated to a POI that is walkable, it has an ACSD of 0 as the trip is not dependent on the road network and schedule. At smaller labeling budgets, the models find it harder to distinguish which zones are made up of walkable POIS which increases the ACSD error term. In Coventry, the issue is compounded as there is a greater proportion of walking only trips in M_g (7.1% vs. 4.3%). Each zone also associates with fewer vaccination centers (6.3 vs. 18.3) so the errors are less diluted by other, non-walking, trips.

Accuracy indicates how well the models can provide each zone with a meaningful accessibility classification with a rule set that considers MAC and ACSD (defined in Section III-D). In Birmingham, an MLP model can provide an accuracy of over 60% at a labeling budget of just 5%, which is a strong result considering there are four possible classes. The accuracies reported for Coventry at lower labeling budgets are inconsistent, and in some cases (e.g., MT) very poor. These poorer results can be explained by the challenges noted above in calculating the ACSD in Coventry.

Finally, we discuss the results for the FIE. The reported errors in this term are low in both Coventry and Birmingham, and remain low even at the lowest labeling budgets. This indicates that the model is able to report the fair distribution of access accurately.

3) Solution Scalability: Table II presents the time costs associated to the naïve approach (e.g., labeling all of M_g) compared to the full end-to-end cost of our solution (feature extraction + labeling $\mathcal{L}$ + SSR learning). Very high costs are associated with full labeling, particularly for Birmingham (e.g., it takes 932 minutes for schools set). With our solution at a 3% labeling budget, the runtime reduces to 36.2 mins, a 96.1% reduction (for which we report a JT error of 3.3 mins, a MAC Corr of 0.84 and accuracy of 0.48). Across all POI types, the mean saving is 96.6% for Birmingham. A higher labeling budget may be preferred in Coventry due to the inconsistent results at lower budget; we consider the performance at 10%. The most intensive query runs in 16.4 minutes, and the mean runtime savings across all POIs times are 91.3%.

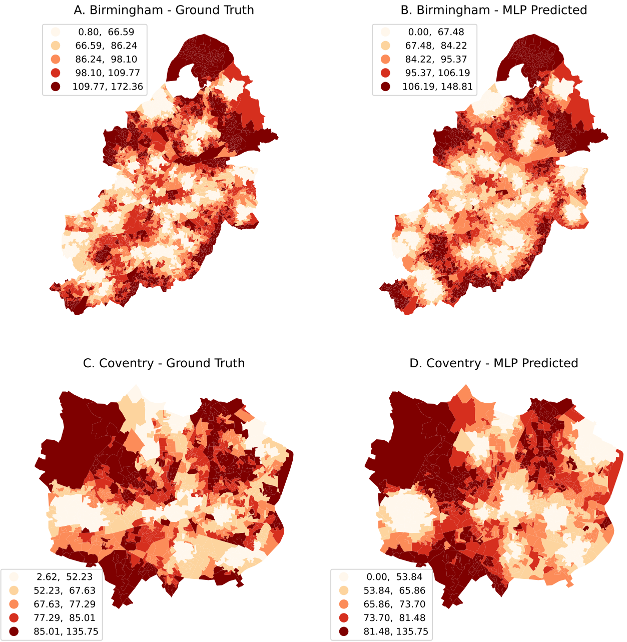


Fig. 5: GAC MAC mapped for vaccination centers.

Fig. 5 maps the predicted MAC for vaccination centers at for Birmingham ($\beta = 3\%$) and Coventry (10%). The results shows that using SSR accurately captures accessibility patterns even with low labeling budgets.

VI. Conclusion

In this work, we have studied the problem of efficiently and accurately processing dynamic AQs by efficiently computing the underlying data to answer such queries. As the spatio-temporal granularities used in accessibility analysis become finer, the computational load to compute the access costs for the full TODAM becomes more impractical. We have defined the challenges and introduced a method to dynamically process AQs for a given set of POIs and a specified time interval, that integrates the gravity model into the construction of the TODAM, and uses semi-supervised regression to significantly reduce the SPQ workload. Our solution dynamically generates feature representations from pre-computed data types, called transit-hop trees, which describe the connectivity in G between an origin and destination. We have also analyzed how fairly access costs are distributed using a system-level measure. In the experiments, we show that the accuracy of measuring accessibility can be maintained for both journey time and the generalized access costs at low labeling budgets

TABLE II: Table showing run time of naive solution compared to SSR solution and percentage savings.

City	**Birmingham**													**Coventry**													
POI	**Label**	**Solution Cost (Mins)**						**Percentage Saving**						**Label**	**Solution Cost (Mins)**						**Percentage Saving**					
Type	**Cost**	**3**	**5**	**7**	**10**	**20**	**30**	**3**	**5**	**7**	**10**	**20**	**30**	**Cost**	**3**	**5**	**7**	**10**	**20**	**30**	**3**	**5**	**7**	**10**	**20**	**30**
School	932.2	36.2	40.9	52.2	75.8	139.1	197.8	96.1	95.6	94.4	91.9	85.1	78.8	190.9	5.1	7.8	10.0	16.3	29.9	43.4	97.3	95.9	94.8	91.5	84.4	77.2
Hospital	876.3	24.7	40.2	51.2	71.0	131.8	191.3	97.2	95.4	94.2	91.9	85.0	78.2	33.1	1.3	1.8	2.2	3.0	5.2	7.5	96.2	94.5	93.2	91.0	84.2	77.3
Vax Center	901.9	25.7	36.5	59.2	72.4	133.5	191.7	97.2	96.0	93.4	92.0	85.2	78.7	72.4	2.3	3.2	4.3	5.7	11.2	15.9	96.8	95.5	94.1	92.1	84.5	78.0
Job Center	305.4	12.4	16.0	22.2	28.6	48.8	69.2	95.9	94.8	92.7	90.6	84.0	77.3	30.3	1.2	1.6	2.0	2.8	4.8	7.1	95.9	94.6	93.2	90.8	84.1	76.7

(3%), and that the fairness index is accurately predicted. This reduces processing times by 97%, allowing dynamic AQs to be answered in minutes rather than hours. Future studies can focus on developing the model and feature representation to better capture walking only trips which drives a low ACSD correlation, as well as investigating the possible use of active learning to further improve the SSR models.

References

[1] S. Kaplan, D. Popoks, C. G. Prato, and A. A. Ceder, "Using connectivity for measuring equity in transit provision," *Journal of Transport Geography*, vol. 37, pp. 82–92, 2014.

[2] S. Farber, M. Z. Morang, and M. J. Widener, "Temporal variability in transit-based accessibility to supermarkets," *Applied Geography*, vol. 53, pp. 149–159, 2014.

[3] A. Owen and D. M. Levinson, "Modeling the commute mode share of transit using continuous accessibility to jobs," *Transportation research part A: policy and practice*, vol. 74, pp. 110–122, 2015.

[4] S. K. Fayyaz, X. C. Liu, and R. J. Porter, "Dynamic transit accessibility and transit gap causality analysis," *Journal of Transport Geography*, vol. 59, pp. 27–39, 2017.

[5] J. P. Bocarejo S and D. R. Oviedo H, "Transport accessibility and social inequities: a tool for identification of mobility needs and evaluation of transport investments," *Journal of transport geography*, vol. 24, pp. 142–154, 2012.

[6] C. Koopmans, W. Groot, P. Warffemius, J. A. Annema, and S. Hoogendoorn-Lanser, "Measuring generalised transport costs as an indicator of accessibility changes over time," *Transport Policy*, vol. 29, pp. 154–159, 2013.

[7] A. Legrain, R. Buliung, and A. M. El-Geneidy, "Travelling fair: Targeting equitable transit by understanding job location, sectorial concentration, and transit use among low-wage workers," *Journal of transport geography*, vol. 53, pp. 1–11, 2016.

[8] A. El-Geneidy, D. Levinson, E. Diab, G. Boisjoly, D. Verbich, and C. Loong, "The cost of equity: Assessing transit accessibility and social disparity using total travel cost," *Transportation Research Part A: Policy and Practice*, vol. 91, pp. 302–316, 2016.

[9] J. Allen and S. Farber, "Planning transport for social inclusion: An accessibility-activity participation approach," *Transportation Research Part D: Transport and Environment*, vol. 78, p. 102212, 2020.

[10] A. Marwal and E. Silva, "Literature review of accessibility measures and models used in land use and transportation planning in last 5 years," *Journal of Geographical Sciences*, vol. 32, no. 3, pp. 560–584, 2022.

[11] A. Páez, D. M. Scott, and C. Morency, "Measuring accessibility: positive and normative implementations of various accessibility indicators," *Journal of Transport Geography*, vol. 25, pp. 141–153, 2012.

[12] K. T. Geurs and B. Van Wee, "Accessibility evaluation of land-use and transport strategies: review and research directions," *Journal of Transport geography*, vol. 12, no. 2, pp. 127–140, 2004.

[13] R. Kujala, C. Weckström, M. N. Mladenović, and J. Saramäki, "Travel times and transfers in public transport: Comprehensive accessibility analysis based on pareto-optimal journeys," *Computers, Environment and Urban Systems*, vol. 67, pp. 41–54, 2018.

[14] D. B. Tomasiello, M. Giannotti, R. Arbex, and C. Davis, "Multi-temporal transport network models for accessibility studies," *Transactions in GIS*, vol. 23, no. 2, pp. 203–223, 2019.

[15] S. Farber and L. Fu, "Dynamic public transit accessibility using travel time cubes: Comparing the effects of infrastructure (dis) investments over time," *Computers, Environment and Urban Systems*, vol. 62, pp. 30–40, 2017.

[16] Y. Jiang, D. Guo, Z. Li, and M. E. Hodgson, "A novel big data approach to measure and visualize urban accessibility," *Computational urban science*, vol. 1, no. 1, pp. 1–15, 2021.

[17] T. Li, P. Meredith-Karam, H. Kong, A. Stewart, J. P. Attanucci, and J. Zhao, "Comparison of door-to-door transit travel time estimation using schedules, real-time vehicle arrivals, and smartcard inference methods," *Transportation Research Record*, vol. 2675, no. 11, pp. 1003–1014, 2021.

[18] E. Ben-Elia and I. Benenson, "A spatially-explicit method for analyzing the equity of transit commuters' accessibility," *Transportation Research Part A: Policy and Practice*, vol. 120, pp. 31–42, 2019.

[19] J. Tu, J. Cheng, and L. Han, "Big data computation for workshop-based planning support," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 1510–1514.

[20] S. K. Fayyaz S, X. C. Liu, and G. Zhang, "An efficient general transit feed specification (gtfs) enabled algorithm for dynamic transit accessibility analysis," *PloS one*, vol. 12, no. 10, p. e0185333, 2017.

[21] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1695–1704.

[22] A. E. Adewale and A. Hadachi, "Neural networks model for travel time prediction based on odtravel time matrix," *arXiv preprint arXiv:2004.04030*, 2020.

[23] F. Zúñiga, J. C. Muñoz, and R. Giesen, "Estimation and prediction of dynamic matrix travel on a public transport corridor using historical data and real-time information," *Public Transport*, vol. 13, no. 1, pp. 59–80, 2021.

[24] H. Yuan, G. Li, Z. Bao, and L. Feng, "Effective travel time estimation: When historical trajectories over road networks matter," in *Proceedings of the 2020 acm sigmod international conference on management of data*, 2020, pp. 2135–2149.

[25] D. Liu, M.-P. Kwan, and Z. Kan, "Assessing job-access inequity for transit-based workers across space and race with the palma ratio," *Urban Research & Practice*, pp. 1–27, 2021.

[26] R. Yang, Y. Liu, Y. Liu, H. Liu, and W. Gan, "Comprehensive public transport service accessibility index—a new approach based on degree centrality and gravity model," *Sustainability*, vol. 11, no. 20, p. 5634, 2019.

[27] W. G. Hansen, "How accessibility shapes land use," *Journal of the American Institute of planners*, vol. 25, no. 2, pp. 73–76, 1959.

[28] J. Kim and B. Lee, "More than travel time: New accessibility index capturing the connectivity of transit services," *Journal of Transport Geography*, vol. 78, pp. 8–18, 2019.

[29] *TAG UNIT M3.2: Public Transport Assignment*. Department for Transport, 2013.

[30] R. K. Jain, D.-M. W. Chiu, W. R. Hawe *et al.*, "A quantitative measure of fairness and discrimination," *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, vol. 21, 1984.

[31] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[32] Z.-H. Zhou, M. Li *et al.*, "Semi-supervised regression with co-training." in *IJCAI*, vol. 5, 2005, pp. 908–913.

[33] N. Jean, S. M. Xie, and S. Ermon, "Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance," *Advances in Neural Information Processing Systems*, vol. 31, 2018.