



Developing a novel approach in estimating urban commute traffic by integrating community detection and hypergraph representation learning

Yuhuan Li^{a,b}, Shaowu Cheng^a, Yuxiang Feng^{b,*}, Yaping Zhang^{a,*}, Panagiotis Angeloudis^b, Mohammed Quddus^b, Washington Yotto Ochieng^b

^a School of Transportation Science and Engineering, Harbin Institute of Technology, Harbin, 150090, China

^b Centre for Transport Engineering and Modelling, Department of Civil and Environmental Engineering, Imperial College London, London, SW7 2AZ, UK

ARTICLE INFO

Keywords:

Commute prediction
Hypergraph learning
Graph neural network
Community detection

ABSTRACT

The efficiency of urban traffic management and congestion alleviation relies heavily on accurate forecasting of Origin-Destination (O-D) demand matrices. Existing models primarily focus on estimating O-D demand for various travel purposes throughout the day, which is characterised by its pulsating nature. However, these models often compromise the precision of peak-hour forecasts, leading to unreliable dynamic traffic control and challenges in effectively reducing peak-hour congestion. To tackle this challenge, this paper proposes a novel method for predicting commuting O-D demand matrices. Our method employs community detection algorithms on road networks to precisely partition commute O-D regions, incorporating Points of Interest (POIs). We also present a spatio-temporal dynamic weighted hypergraph model that leverages these partitioned regions, time characteristics from observed O-D trips, and meteorological data to improve forecasting. Comparative analyses with contemporary models and ablation studies indicate our method significantly enhances prediction accuracy, by approximately 5%. These findings imply that the proposed method more effectively encompasses the varied characteristics of commuting during peak hours, thereby providing more accurate demand matrices for urban traffic management.

1. Introduction

Origin-Destination (O-D) demand matrices are vital for urban adaptive traffic control (Ashok & Ben-Akiva, 2002). However, these matrices cannot be directly obtained due to daily traffic flow fluctuations influenced by various factors such as weekdays, weather conditions, and events (Sun et al., 2017). Consequently, estimating urban O-D demand matrices using observed traffic data has garnered significant attention in recent years. Previous and current research has accomplished the estimation of O-D demand matrices without differentiating travel purposes (Ke et al., 2021; Liang et al., 2023; Zhang et al., 2021; Zou et al., 2022).

For instance, Liang et al. (2023) used two taxi order datasets from Shenzhen City and New York City, along with the existing traffic zones of these cities, to estimate the traffic demand over 1–2 weeks in both cities. This method lacks clear differentiation of specific travel purposes. Instead, it employs predictive models trained on continuous observational data, aiming to forecast O-D matrices over a subsequent extended period.

In fact, daily travel patterns encompass multiple purposes at different time of the day, such as commuting during peak hours, tourism and shopping around midday, and leisure activities in the evening (Gao et al., 2022; Soares et al., 2019). These varying purposes naturally have distinct origins and destinations (Guo et al., 2019; Jiang et al., 2022), like commuting typically involving trips between residences and workplaces. Using the same O-D partitions for all purposes may lead to significant spatial feature discrepancies, potentially deteriorating prediction accuracy. Moreover, traffic volumes exhibit significant fluctuations throughout the day, characterised by peak-hour surges, midnight troughs, and comparatively stable flows during other periods (Liang et al., 2023; Ye et al., 2022; Zou et al., 2022). Methods that estimate the matrices with the objective of minimising overall error may consequently diminish the precision of estimations during peak hours.

Therefore, while current methods have addressed overall O-D demand matrix estimation and forecasting, the estimation of demand matrices for specific travel purposes has received scant attention. To

* Corresponding authors.

E-mail addresses: yuhuan.li22@imperial.ac.uk (Y. Li), csw_h@hit.edu.cn (S. Cheng), y.feng19@imperial.ac.uk (Y. Feng), zxlt0905@163.com (Y. Zhang), p.angeloudis@imperial.ac.uk (P. Angeloudis), m.quddus@imperial.ac.uk (M. Quddus), w.ochieng@imperial.ac.uk (W.Y. Ochieng).

<https://doi.org/10.1016/j.eswa.2024.123790>

Received 18 December 2023; Received in revised form 30 January 2024; Accepted 18 March 2024

Available online 21 March 2024

0957-4174/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

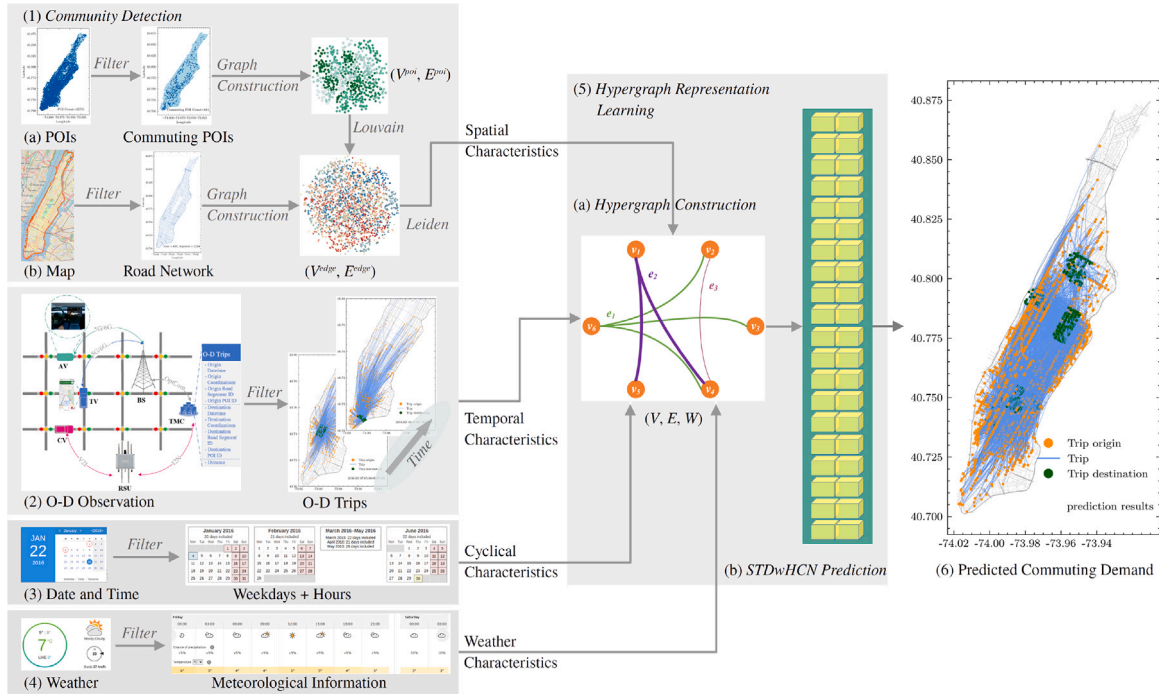


Fig. 1. Framework for urban commute traffic prediction. (1) Process of O-D region division based on community detection, (2) Processing of O-D trip observation data, (3) Procedure for handling date and time information, (4) Process of weather data, (5) Hypergraph representation learning procedure.

bridge this gap, this paper focuses on estimating urban commuting demand matrices during peak hours, employing O-D observational data, weather information, and temporal characteristics of workdays and peak hours. The methodology underpinning this prediction is detailed in Fig. 1.

The primary contributions of this paper are summarised as follows:

- Based on community detection and hypergraph representation learning, we propose a framework to estimate urban commuting traffic during peak hours, which could provide a more accurate O-D demand matrix for dynamic traffic control.
- Leveraging the locations of commute-related POIs, we use community detection algorithms to refine O-D partitioning on road networks, providing more accurate spatial features to the proposed commuting estimation model.
- A novel model is developed for commuting O-D demand prediction based on hypergraph representation learning, which effectively integrates and captures heterogeneous characteristics.

The remainder of the paper is structured as follows: Section 2 offers a brief review of relevant literature pertaining to community detection algorithms and traffic prediction models. Section 3 provides a detailed description of the methodologies employed. The results and discussion are presented in Section 4. Finally, Section 5 concludes the paper and proposes directions for further research.

2. Related work

In the context of this paper, we have conducted a comprehensive review of the literature, focusing on community detection methods and O-D demand prediction models.

2.1. Community detection

Community detection algorithms have been applied across various fields. The initial exploration of this concept began in the 1980s (Holland et al., 1983). However, it gained broader recognition in physics

about two decades ago, largely due to Girvan and Newman's pioneering efforts (Newman & Girvan, 2004). Nowadays, a large number of community detection methods have been previously developed, which can be divided into different categories. As we are conducting community detection on graphs, we categorise the models into directed and undirected graphs.

Directed graphical models for community detection began with the introduction of the stochastic block model (SBM) in 1983, a concept initially used for social networks (Holland et al., 1983). This field has since evolved, giving rise to a variety of models aimed at improving network structure analysis. An important contribution in this area is by Jin et al. (2020), who explored the relevance and expected degree of each node in a community, applying a Bayesian framework to determine community numbers. This study particularly focused on the identification of overlapping communities in networks.

However, these graphical models often overlook varied community patterns and struggle with noisy or sparse topological data. To overcome these shortcomings and improve community detection in intricate networks such as those in transportation, the research focus has shifted towards models for undirected graph community detection (Blondel et al., 2008; Ghosh et al., 2018; He et al., 2018; Nowozin, 2010; Zeng & Yu, 2018).

Initially, undirected graphical models found extensive application. For example, Nowozin (2010) employed discrete undirected graphical models grounded in Markov Random Field (MRF) for data clustering, facilitating structured learning and prediction tasks. More recently, He et al. (2018) advanced this approach with their development of Network-specific MRF (NetMRF). This adaptation is effective in community detection within large-scale networks, optimally leveraging unique network topology data.

The Louvain algorithm, introduced by Blondel et al. (2008), represents another effective method for detecting communities in undirected graphs. Renowned for its efficiency in identifying partitions with high modularity in expansive networks, this algorithm has become a popular tool among experts. Efforts to boost its efficiency further have led to various parallelisation attempts of the Louvain heuristic. For example, Ghosh et al. (2018) developed a distributed memory variant of the Louvain algorithm (DLA) using MPI+OpenMP, achieving

Table 1
Related work to O-D demand prediction.

Method	Authors	Date	Required features	Dataset (Techniques)
ST-ED-RMGC	Ke et al. (2021)	2021	- Region graph - Previous observation	New York City Taxi (GPS on TVs)
DNEAT	Zhang et al. (2021)	2021	- Region graph - Previous observation	Kunshan Dataset (Mobile Phones)
ESTGCN or VSTGCN	Liu et al. (2022)	2022	- Region graph - Previous observation	- Chengdu, Didi (Navigation Apps on TVs) - New York City Taxi (GPS on TVs)
ST-GDL	Zou et al. (2022)	2022	- Region graph - Previous observation	- Haikou, Didi (Navigation Apps on TVs) - New York City Taxi (GPS on TVs)
LSTM + DLNa	Pamula and Żochowska (2023)	2023	- Road network model - Prior O-D matrix - Previous observation	Traffic volume in Gliwice (Detectors)
DAGNN	Zhang and Xiao (2023)	2023	- Region graph - Previous observation - POIs	- Chengdu, Didi (Navigation Apps on TVs) - New York City Taxi (GPS on TVs)
MARGAT	Liang et al. (2023)	2023	- Region graph - Previous observation - POIs	- Shenzhen Taxi (Navigation Apps on TVs) - New York City Taxi (GPS on TVs)

approximately a 7-fold increase in speed without sacrificing quality. Additionally, Zeng and Yu (2018) proposed a Scalable Distributed Louvain Algorithm (SDLA) designed for scalability and precision in community detection within large-scale graphs.

However, the Louvain algorithm can yield poorly connected communities and may even lead to disconnected communities through multiple iterations. To address this problem, Traag et al. (2019) developed a Leiden algorithm. Tested in real-world networks, partitioning results indicate that this algorithm not only ensures community connections but also surpasses the Louvain algorithm in terms of convergence speed.

In summary, this literature review points out established theories and methods for detecting communities in graphs. Yet, these methods have not been widely used for partitioning O-D regions. Despite this, algorithms for community detection in undirected graphs are quite suitable for such tasks. They efficiently deal with varying densities, intersecting communities, and time-related changes in the O-D data. Therefore, this paper employs community detection algorithms to partition O-D regions.

However, traditional approaches use major roads as boundaries for partitioning O-D areas (Liang et al., 2023), similar to the Manhattan taxi zones. In fact, effective traffic control and management primarily focus on road segments rather than these broader zones. As a result, traditional O-D zones contain redundant information that may not be useful for traffic control, such as details of buildings or internal community roads. Moreover, existing O-D zones are consistent for all types of traffic demand (Ke et al., 2021; Wang et al., 2022), which leads to inaccuracies in estimating specific demands, such as commuting. This is because different types of travel have distinct origins and destinations. For example, commuting typically involves travel from a residence to a workplace, whereas shopping trips often end at destinations like shopping centres.

To address these limitations, our method utilises a community detection algorithm to partition O-D areas on the road network, ensuring the partitioning result is more practical for real traffic management. Furthermore, POIs pertinent to commuting are considered when partitioning O-D areas. By focusing on the proximity between these commuting POIs and road segments, we aim to achieve a more accurate partition for commuting O-D areas, thereby improving the precision of subsequent estimates for commuting O-D demand.

2.2. O-D demand prediction

Recently, there has been an increasing focus on creating sophisticated models to forecast dynamic O-D demand patterns. A detailed summary of these models is provided in Table 1.

One of notable models is the Spatio-Temporal Encoder-Decoder Residual Multi-Graph Convolutional Network (ST-ED-RMGC). This model effectively employs graph-based techniques to understand spatial and temporal dependencies and has been successful in forecasting ride-sourcing demand across various locations (Ke et al., 2021). Another innovative approach is the Dynamic Node-Edge Attention Network (DNEAT) (Zhang et al., 2021), which utilises dynamic O-D graphs to track changes in node topologies over time. DNEAT integrates Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) mechanisms with Graph Convolutional Networks (GCN) to effectively process sparse demand data.

Liu et al. (2022) developed two models based on Spatio-Temporal Graph Convolutional Networks (STGCN) for predicting dynamic O-D flows, leveraging graphs derived from mobile phone data. The first model concentrates on updating edge features in large-scale graphs, while the second assesses relationships among O-D flows through a temporally similar adjacency matrix. Additionally, Zou et al. (2022) introduced a Graph Deep Learning model for long-term O-D prediction (ST-GDL), combining Convolutional Neural Network (CNN) and graph convolution techniques to discern both short-term and long-term trends.

Zhang and Xiao (2023) put forward the Dynamic Autostructuring Graph Neural Network (DAGNN) to predict ride-hailing demand between specific origins and destinations, showing remarkable ability to learn time-varying graph contexts. Liang et al. (2023) created the Multi-task Adaptive Recurrent Graph Attention Network (MARGAT) for precise regional travel demand forecasting. This model synergies prior knowledge-driven graph learning with a recurrent graph attention mechanism, effectively capturing dynamic spatio-temporal dependencies.

In summary, while the current literature underscores the potential of deep learning models in forecasting dynamic O-D demand, these models predominantly utilise conventional graphs that depict point-to-point relationships, such as journeys between individual O-D regions. However, this method has limitations in fully encapsulating the complexity of trips spanning multiple regions, for instance, a journey commencing in one area, traversing through another, and completing in a third. Therefore, traditional graph models find it challenging to comprehensively represent these multi-node relational trips.

In contrast, hypergraphs provide a substantial advantage through the use of hyperedges that can connect multiple nodes (Antelmi et al., 2024; Viñas et al., 2023; Wang & Zhu, 2024; Yan et al., 2023), offering an exhaustive portrayal of trips that cover several regions. Consequently, we propose a novel spatio-temporal dynamic weighted hypergraph model, designed to capture and learn the intricate spatial and

temporal features present in O-D historical data, thereby providing a more accurate and holistic understanding of urban traffic patterns.

Moreover, while the majority of existing research focuses on macro-level O-D demand prediction (Liang et al., 2023; Pamula & Żochowska, 2023; Zhang & Xiao, 2023; Zou et al., 2022), there exists a notable gap in forecasting specific demands, especially in the realm of daily commuting O-D demand, which is most prevalent during peak hours on weekdays. This aspect presents a greater challenge due to the high frequency of commuting trips. Therefore, our study aims to estimate O-D demand during these peak periods, utilising historical O-D data and the proposed commuting regions as key spatio-temporal features.

Additionally, many studies overlook the impact of weather conditions on commuting choices. Commuters might choose public transportation instead of driving under adverse weather conditions, such as heavy rain or snow. Consequently, this study incorporates meteorological data from historical periods to investigate the influence of weather on O-D travel patterns.

3. Methodology

3.1. Commute region division using community detection

3.1.1. Problem description

To accurately predict urban commuting travel demand, it is essential to first establish commuting O-D regions as a spatial feature for input into forecasting models. Commuting generally entails travel between residential areas and workplaces. Typically, in urban areas, there is a noticeable clustering of these locations, such as financial centres and industrial areas. This clustering phenomenon is fundamental to understanding commuting patterns. To ensure that our O-D partitioning considers the aggregation of commuting-related places (POIs), we have developed a commuting O-D partitioning algorithm, as detailed in Algorithm 1.

Initially, a graph structure, denoted as G^{poi} , is constructed, incorporating both Euclidean and path distances among POIs. Subsequently, these POIs are clustered using the Louvain algorithm on this graph structure. Subsequently, a second graph, G^{edge} , is developed to partition all road segments within the network. During this process, the clustering results of the POIs are integrated as weighting factors between the road segment nodes in G^{edge} . This integration is crucial as it transfers the spatial clustering features of commuting-related POIs into the road network partitioning, thereby enhancing the relevance and accuracy of the partitioning process. To complete the task of road network partitioning, the Leiden algorithm is employed for community detection, partitioning each road segment of the network into separate communities.

3.1.2. Commuting POI partition

The initial phase of Algorithm 1 for partitioning commuting POIs involves three main steps: data preprocessing, graph construction, and POI clustering with the Louvain algorithm.

(1) Data Preprocessing

As depicted in Fig. 1 (1), commuting POIs P are filtered from all urban POIs based on the POI type p_i^f . The filtered POIs, hereafter simply referred to as POIs, represent primary locations for commuting activities. Subsequently, utilising the geographic coordinates of each POI p_i^{geo} and the road segment geometry r_m^{geo} , each POI p_i is associated with its nearest road segment r_m . This association is denoted by $p_i^{r,id}$, which identifies the nearest road segment to p_i .

Next, using the geographic coordinates p_i^{geo} , the Euclidean distance matrix D^e for every pair of POIs is computed as,

$$D^e = [d_{ij}^e] \quad \text{where} \quad d_{ij}^e = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

where, (x_i, y_i) and (x_j, y_j) are the geographic coordinates of POI p_i and p_j , respectively. d_{ij}^e represents the Euclidean distance between p_i and p_j .

Algorithm 1 Commuting Region Division

Input:

- Commuting POIs P , Road network topology R
- Parameters: k , $radius$, n , $(\alpha_e, \alpha_j, \alpha_s, \alpha_n)$, r_{poi} , L , r_{edge} , $(\beta_b, \beta_p, \beta_s, \beta_n, \beta_d)$

Output: Road segment cluster $R^{cluster}$

```

1: Initialise POI clustering parameters:  $k$ ,  $radius$ ,  $n$ ,  $r_{poi}$ ,  $(\alpha_e, \alpha_j, \alpha_s, \alpha_n)$ 
2: for  $i = 0$  to number of POIs do
3:   for  $j = 0$  to number of POIs do
4:     Euclidean distance  $d_{ij}^e \leftarrow (p_i^{geo}, p_j^{geo})$ 
5:     Path distance  $d_{ij}^p$ , and number of road segments  $f_{ij}^p \leftarrow (p_i^{r,id}, p_j^{r,id})$ 
6:   end for
7: end for
8: for  $i = 0$  to number of POIs do
9:   Graph  $G^{poi} = (V^{poi}, E^{poi}) \leftarrow D^e = [d_{ij}^e]$ ,  $\alpha_e$ ,  $radius$ 
10: end for
11: for  $i = 0$  to number of POIs do
12:    $k$  nearest POIs  $N_i \leftarrow$  Path distances  $d_{ij}^p$ 
13:   for  $j = 0$  to  $k$  do
14:     Jaccard similarity  $w_j^{poi} \leftarrow N_i$  and  $N_j$ 
15:     Compute edge weight  $w_s^{poi}$ ,  $w_n^{poi}$ 
16:     Update edge weight with  $f_{ij}^p$ ,  $\alpha_j$ ,  $\alpha_s$ ,  $\alpha_n$ 
17:   end for
18: end for
19:  $\{p_i^{cluster}\} \leftarrow$  Apply Louvain on  $G^{poi}$  with  $[d_{ij}^p]$ ,  $r_{poi}$ 
20: Initialise road segment clustering parameters:  $L$ ,  $r_{edge}$ ,  $(\beta_b, \beta_p, \beta_s, \beta_n, \beta_d)$ 
21: for  $m = 0$  to number of road segments do
22:   for  $n = 0$  to number of road segments do
23:     Graph  $G^{edge} = (V^{edge}, E^{edge}) \leftarrow (r_m^{SN}, r_m^{EN}), (r_n^{SN}, r_n^{EN}), w_b^{edge}$ 
24:     Compute weight  $w_p^{edge}$ ,  $w_s^{edge}$ ,  $w_n^{edge}$ ,  $w_d^{edge}$ 
25:     Update weight with  $p_i^{cluster}$ ,  $L$ ,  $(\beta_p, \beta_s, \beta_n, \beta_d)$ 
26:   end for
27: end for
28:  $\{r_m^{cluster}\} \leftarrow$  Apply Leiden in RBConfigurationVertexPartition mode on  $G^{edge}$  with  $r_{edge}$ 
29: return  $R^{cluster} = \{r_0^{cluster}, r_1^{cluster}, \dots, r_m^{cluster}\}$ 

```

Utilising $p_i^{r,id}$ and applying Dijkstra's algorithm (Dijkstra, 2022) to the road network, the path distance matrix between POIs $D^p = [d_{ij}^p]$ and the matrix of the total number of road segments involved in each path $F^p = [f_{ij}^p]$ can be computed.

(2) Graph Construction

To cluster POIs, we construct a graph $G^{poi} = (V^{poi}, E^{poi})$, where POIs are the vertices and edge have composite weights w^{poi} . The edge weight w^{poi} is composed of the Euclidean distance weight w_e^{poi} , the same road segment weight w_s^{poi} , the limiting weight w_n^{poi} based on the number of road segments in the path, and the Jaccard similarity weight w_j^{poi} . The computation of these weights is,

$$w^{poi} = \alpha_e \cdot w_e^{poi} + \alpha_s \cdot w_s^{poi} + \alpha_n \cdot w_n^{poi} + \alpha_j \cdot w_j^{poi} \quad (2)$$

where α_e , α_s , α_n , α_j are the weight coefficients. Specific weight calculations are as follows:

- w_e^{poi} is set to 1 if the Euclidean distance d_{ij}^e is less than $radius$, otherwise 0.
- w_s^{poi} is 1 if POIs p_i and p_j are on the same road segment (i.e., $p_i^{r,id} = p_j^{r,id}$), otherwise 0.
- w_n^{poi} is 1 if the number of road segments in the path f_{ij}^p is less than n road segments, otherwise 0.

- w_j^{poi} is computed as,

$$w_j^{poi} = \frac{|N_i \cap N_j|}{|N_i \cup N_j|} \quad (3)$$

where N_i and N_j denote the sets of the k -nearest POIs to p_i and p_j respectively, determined by their path distance.

(3) POI Clustering with the Louvain Algorithm

The partitioning of POIs into separate communities is executed through the application of the Louvain algorithm (Blondel et al., 2008) on the graph G^{poi} . The Louvain algorithm's fundamental principle involves optimising community structures by maximising the modularity Q^{poi} . This modularity is a metric that evaluates the concentration of links inside communities relative to those between communities. The calculation of modularity Q^{poi} can be described as,

$$Q^{poi} = \frac{1}{2 \sum_0^i w_i^{poi}} \sum_{i,j} \left(w_{ij}^{poi} - r_{poi} \frac{k_i k_j}{2 \sum_0^i w_i^{poi}} \right) \delta(c_i, c_j) \quad (4)$$

where,

- w_{ij}^{poi} is the weight of the edge between POIs p_i and p_j , $w_{ij}^{poi} = 0$ if no edge exists;
- r_{poi} is a resolution parameter, adjusting the modularity optimisation scale.
- k_i and k_j are the degrees (sum of the weights of the edges connected to them) of POI p_i and p_j , respectively;
- $\sum_0^i w_i^{poi}$ is the sum of all edges' weights in the graph;
- c_i and c_j represent the community assignments of POIs p_i and p_j , respectively;
- $\delta(c_i, c_j)$ is an indicator function that is 1 when $c_i = c_j$ and 0 otherwise.

Following the computation of modularity Q^{poi} , the modularity gain ΔQ_i for each POI p_i is defined. This measures the change in modularity resulting from moving p_i to a different community c . The gain ΔQ_i is calculated using,

$$\Delta Q_i = \left[\frac{\sum_c w_c + 2k_{i,c}}{\sum_0^i w_i^{poi}} - \left(\frac{\sum_c^d + k_i}{\sum_0^i w_i^{poi}} \right)^2 \right] - \left[\frac{\sum_c w_c}{\sum_0^i w_i^{poi}} - \left(\frac{\sum_c^d}{\sum_0^i w_i^{poi}} \right)^2 - \left(\frac{k_i}{\sum_0^i w_i^{poi}} \right)^2 \right] \quad (5)$$

where,

- $k_{i,c}$ is the sum of the weights of the edges from POI p_i to all POIs in community c ;
- $\sum_c^w$ is the sum of all edges' weights within community c ;
- $\sum_c^d$ is the sum of the degrees of all POIs in community c .

During the local optimisation phase, ΔQ_i is computed for each POI p_i by considering its transfer to adjacent communities that maximise ΔQ_i . Subsequently, all POIs within each community c are merged into a single 'super-POI', followed by a recalculation of the edge weights. This iterative process is repeated until there is no further improvement in the modularity Q^{poi} . It is important to note that due to the stochastic nature of the Louvain algorithm, the final partitioning of POIs into communities may exhibit slight variations across different runs. However, each resulting partitioning is considered valid and effectively contributes to the robust clustering of the POIs.

3.1.3. Road segment partition

The second half of Algorithm 1 outlines the process of road segment partitioning, which can be distilled into three main steps: data preprocessing, graph construction, and clustering of road segments using the Leiden algorithm. This approach hinges on the connectivity of road segments and the commuting POI clusters previously identified,

denoted as $P^{cluster} = \{p_0^{cluster}, p_1^{cluster}, \dots, p_i^{cluster}\}$. This methodology effectively captures accurate static spatial features, which are crucial for enhancing the precision of subsequent O-D demand prediction models.

(1) Data Preprocessing

As shown in Section 1 of Fig. 1, the first step in road segment partition is extracting road network data from a city map. This process encompasses the collection of detailed attributes for each road segment, such as the geometry information r_m^{geo} , speed limits r_m^{speed} , number of lanes r_m^{lane} , the length of the road segment r_m^{length} , and identifiers for the start junction r_m^{SN} and end junction r_m^{EN} of each segment.

(2) Graph Construction

For the road segment analysis, we develop a separate graph $G^{edge} = (V^{edge}, E^{edge})$, where each road segment is represented as a vertex. The edges in this graph denotes the connections between road segments, and they are assigned composite weights w^{edge} . These weights are an amalgamation of various factors: the basic weight w_b^{edge} , a weight for connecting road segments with the same type of POIs w_p^{edge} , a weight for segments sharing the same speed limit w_s^{edge} , a weight for segments with identical lane counts w_n^{edge} , and a weight for segments where the length disparity is smaller than the minimum length of any road segment in the network w_d^{edge} . The calculation of these weights can be denoted as,

$$w^{edge} = \beta_b \cdot w_b^{edge} + \beta_p \cdot w_p^{edge} + \beta_s \cdot w_s^{edge} + \beta_n \cdot w_n^{edge} + \beta_d \cdot w_d^{edge} \quad (6)$$

where $\beta_b, \beta_p, \beta_s, \beta_n, \beta_d$ are the weight coefficients. Specific weight calculations are as follows,

- w_p^{edge} for road segments associated with the same type of POIs is set to 1 if they belong to the same cluster, otherwise 0.
- w_s^{edge} is 1 for road segments sharing the same speed limit, else 0.
- w_n^{edge} is assigned a value of 1 for road segments with an equal number of lanes, else 0.
- w_d^{edge} takes the value 1 if the length difference between road segments is less than the minimum length L , otherwise 0. Where, $L = \min\{r_0^{length}, r_1^{length}, \dots, r_m^{length}\}$.
- For the basic weight w_b^{edge} ,

$$w_b^{edge} = \begin{cases} 1, & \text{if } \{r_m^{SN}, r_m^{EN}\} \cap \{r_n^{SN}, r_n^{EN}\} \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

(3) Road Segment Clustering with the Leiden Algorithm

To cluster road segments into communities, the Leiden algorithm (Traag et al., 2019) in its RBConfigurationVertexPartition mode is applied to the graph G^{edge} . This mode is based on Reichardt and Bornholdt's Potts model (Reichardt & Bornholdt, 2006), making it an optimal choice for graphs with positive edge weights. It employs a linear resolution parameter r_{edge} in its calculations. Similar to the Louvain method, this approach focuses on optimising community structures through the maximisation of modularity. The quality function employed for clustering road segments is defined as,

$$Q^{edge} = \sum_{m,n} \left(w_{mn}^{edge} - r_{edge} \frac{k_m k_n}{2 \sum_0^m w_m^{edge}} \right) \delta(c_m, c_n) \quad (8)$$

where,

- w_{mn}^{edge} is the weight of the edge between road segments r_m and r_n , with $w_{mn}^{edge} = 0$ if no edge exists.
- r_{edge} is the resolution parameter, adjusting the modularity optimisation scale.
- k_m and k_n are the degrees of road segments r_m and r_n .
- $\sum_0^m w_m^{edge}$ is the total edge weight in the graph.
- c_m and c_n represent the community assignments of road segments r_m and r_n .
- $\delta(c_m, c_n)$ is an indicator function that is 1 if $c_m = c_n$ and 0 otherwise.

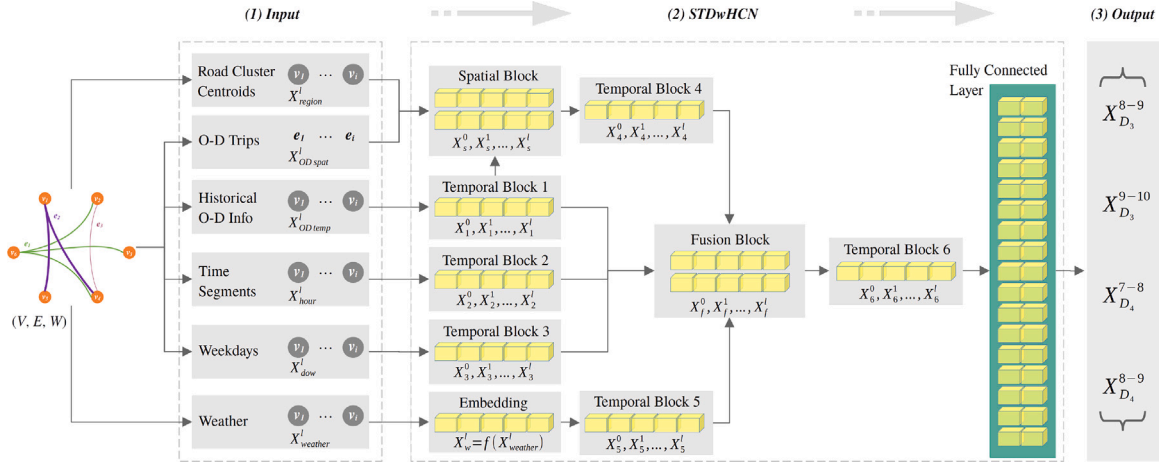


Fig. 2. The architecture of the forecasting model.

The Leiden algorithm iteratively refines the community partitioning by locally optimising the modularity gain for individual road segments, followed by a phase of aggregating segments into larger communities. This iterative process is repeated until there is no substantial increase in the overall modularity, ensuring that the final partitioning accurately and robustly represents the underlying connectivity and structural patterns within the road network.

3.2. Commuting O-D matrix prediction

3.2.1. Problem description

After clustering road segments, trip data are allocated to respective clusters. The number of O-D pairs $[o_t^i, d_t^i]$ is then quantified on an hourly basis. Concurrently, to incorporate the impact of external variables, data including time segment h_t and day of the week dow_t are integrated, employing cyclical encoding to delineate recurring traffic patterns. Meanwhile, throughout the data collection phase, 15 distinct weather conditions were observed from historical data. These conditions are further categorised into two primary groups: one comprising 5 nominal weather types (fair, cloudy, haze, drizzle/rain, and snow) and the other delineating 3 levels of ordinal weather intensity (light/partly, mild, and mostly/heavy). Consequently, this meteorological data is encoded using two separate embedding layers, $E_{cond}(c_t)$ and $E_{int}(i_t)$. The resultant feature can thus be denoted as,

$$F_t^i = [o_t^i, d_t^i, \sin(h_t), \cos(h_t), \sin(dow_t), \cos(dow_t), E_{cond}(c_t), E_{int}(i_t)] \quad (9)$$

Given a time period $t \in \{1, \dots, T_{obs}\}$ and N road segment clusters in the scene, this study aims to predict the future generated and attracted O-D demand in these clusters $\hat{s}_t^i = (\hat{o}_t^i, \hat{d}_t^i), i \in \{1, 2, \dots, N\}$ over a time horizon $\hat{t} \in [T_{obs}+1, \dots, T_{pred}]$ concurrently, based on a set of observations F_t^i . With the extracted data, this model predicts the O-D demand for the future 4 h based on observations over the past 7 h.

3.2.2. Model architecture

The architecture of the proposed model is illustrated in Fig. 2. It consists of 4 temporal blocks, 1 spatial block, 1 fusion block and 1 fully connected layer.

Temporal Convolution. Recurrent Neural Networks (RNNs) are commonly used for temporal feature extraction at individual vertices. However, RNNs encounter higher computational complexity, generally in the order of $O(d^2)$, in contrast to the $O(kd)$ complexity of Convolutional Neural Networks (CNNs). RNNs rely on outputs from previous steps, which can result in delayed responses and extended computation times. In comparison, CNNs efficiently identify temporal dependencies with a more efficient architecture and allow parallel

processing of features due to the independent nature of their convolutional computations. Given the short length of our sequential data (totalling 11 units), the dilation convolution operation, a key aspect of Temporal Convolutional Networks (TCNs), is less beneficial. Therefore, we implement a sequence of three CNNs within our temporal block for effective temporal feature extraction.

$$X^{l+1} = \sigma_t(\theta_1 * X^l + \sigma_t(\theta_2 * X^l)) + \theta_3 * X^l \quad (10)$$

where σ_t is the nonlinear activation function, X is the temporal traffic features and θ is the convolution kernel.

Dynamic Weighted Hypergraph Construction. Unlike a traditional graph $G = (V, E)$, which is composed of vertices $v \in V$ connected in pairs by edges $e \in E$, a hypergraph $G^h = (V, E, W)$ extends this concept by allowing each hyperedge $e \in E$ to connect any number of vertices. The strength of these connections is quantified using a diagonal weight matrix $W \in \mathbb{R}^{I \times I}$. Hypergraphs are typically represented through an incidence matrix $H \in \mathbb{R}^{N \times I}$, which delineates the interactions between N vertices and I hyperedges. In most weighted hypergraph research, the weight matrix W is used to incorporate variable weights (Kang et al., 2022; Ma et al., 2023), whereas the incidence matrix H is generally binary, indicating presence or absence of vertex-edge connections.

$$h = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

In this research, the centroid of each road segment cluster is considered a vertex within a hypergraph, with their common travel routes represented as hyperedges. As O-D trip data are sparse and only contain two vertices, acquiring shared route details is challenging since trips may traverse additional vertices. To address this, we constructed the road network of the Manhattan area and employed the Dijkstra algorithm to recreate travel routes from the O-D data of each trip, thereby gaining insights into all road segment clusters involved. This route data is represented as a binary vector of fixed length, with '1' indicating the inclusion of a vertex in a trip and '0' indicating exclusion. By converting trip information into binary vectors, we then calculate the Hamming distance between these vectors and apply Ward's method of hierarchical clustering. This process segments the trips into clusters, each representing a hyperedge with aggregated trip data. The hypergraph is dynamically updated at each time step to more accurately capture changes in O-D commuting patterns.

However, it is important to acknowledge that the hyperedges in this study, formed through clustering of trip data, may comprise varying numbers of trips per vertex. This variation underscores the differing significance of vertices within a hyperedge. The traditional binary incidence matrix fails to reflect this disparity. Drawing inspiration

from Chitra and Raphael (2019) and Zhu and Segarra (2023), we have innovated an adaptive approach for calculating vertex weights in relation to edge dependence, based on trip frequency and travel distance. This method allows for the spatial aspects to be incorporated, resulting in a more nuanced representation in the incidence matrix, which can be denoted as,

$$h_{ve} = \begin{cases} \alpha \frac{\hat{f}_v}{\hat{d}_e}, & \text{if } v \in e \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where α is a learnable parameter, $\hat{f}_v$ is the normalised trip frequency at vertex v , $\hat{d}_e$ is the normalised trip distance in edge e .

The degree of a vertex v is defined as the summation of all attached hyperedges' weights w_e ,

$$d_v = \sum_{e \in E} w_e h_{ve}, e \in E \quad (13)$$

The degree of a hyperedge e is defined as the number of vertices it connects,

$$d_e = \sum_{v \in V} h_{ve}, e \in E \quad (14)$$

Spatial Convolution. The calculation of the hypergraph Laplacian, denoted as Δ , employs the incidence matrix H_{ve} (elements h_{ve}), the weight matrix W (elements w_e), and the diagonal matrices of vertex degree D_v (elements d_v) and hyperedge degree D_e (elements d_e). This can be described as,

$$\Delta = I - D_v^{-\frac{1}{2}} H_{ve} W D_e^{-1} H_{ve}^T D_v^{-\frac{1}{2}} \quad (15)$$

where Δ is a positive semi-definite matrix with only real eigenvalue.

Building on the spectral hypergraph convolution concept introduced by Feng et al. (2019), the hypergraph convolution layer is defined as,

$$X^{(l+1)} = \sigma_s \left(D_v^{-\frac{1}{2}} H_{ve} W D_e^{-1} H_{ve}^T D_v^{-\frac{1}{2}} X^{(l)} \Theta^{(l)} \right) \quad (16)$$

where σ_s is the nonlinear activation function, $X^{(l)}$ is the matrix of vertex features at the l -th layer.

Fusion Mechanism. After spatial and temporal features are extracted through designated blocks, they are integrated using a fusion process to encapsulate the combined spatio-temporal dependencies in the network. This integration is mathematically represented as,

$$X = \sigma_f \left(\text{concat}(X_{\text{OD_temp}} \cdot \Theta_1, X_{\text{OD_spat}} \cdot \Theta_2, X_{\text{hour}} \cdot \Theta_3, X_{\text{dow}} \cdot \Theta_4, X_{\text{weather}} \cdot \Theta_5) \right) \quad (17)$$

Here, $X_{\text{OD_temp}}$ and $X_{\text{OD_spat}}$ represent the temporal and spatial features extracted from O-D data, respectively. The variables X_{hour} , X_{dow} , and X_{weather} correspond to the temporal features associated with the time segment h , day of the week dow , and weather conditions, which are captured through concatenated embeddings $[E_{\text{cond}}(c), E_{\text{int}}(i)]$. The symbol $\cdot$ signifies matrix multiplication, and $\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5$ are the respective learnable parameters involved in this fusion process, σ_f is the nonlinear activation function.

Loss Function. The training objective of the model is to minimise the Mean Squared Error (MSE) loss. This loss is calculated as the average of the squares of the differences between the predicted and actual O-D demand. The loss function can be represented as,

$$L_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (18)$$

3.3. Evaluation metrics

3.3.1. Commuting O-D region division

In terms of community detection algorithms, modularity is a metric that measures the structure of a network, assessing whether the community structure of a network is significant (Newman & Girvan, 2004). For

graphs G^{poi} and G^{edge} , the modularity is defined as Eq. (4) and Eq. (8), respectively.

For clustering algorithms, Silhouette Coefficient is a useful metric for assessing scatter plot clustering when ground truth labels are unknown (Rousseeuw, 1987). This metric is calculated using the following formula:

$$s = \frac{b - a}{\max(a, b)} \quad (19)$$

where a represents the mean distance between a sample and all other points within the same class, and b denotes the mean distance between a sample and all other points in the nearest cluster different from its own.

3.3.2. O-D demand prediction

Three commonly used evaluation metrics are used in this study to evaluate the prediction accuracy of the proposed method, which are the mean absolute error (MAE), root mean square error (RMSE), and symmetric mean absolute percentage (sMAPE) (Chen et al., 2023; Essien et al., 2021). MAE measures the average absolute difference between the ground truth and the predicted O-D demand, RMSE calculates the square root of the mean squared error, and sMAPE calculates the percentage difference between the predicted and actual O-D demand.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (20)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (21)$$

$$sMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \quad (22)$$

4. Results and discussion

This section provides an in-depth evaluation of the proposed methodology across three key aspects of this study. First, we begin by exploring the test scenario and the data preprocessing steps involved. Second, we focus on the proposed approach for partitioning commuting regions and offer a comparative analysis with alternative methods. The last part of this evaluation delves into the process of predicting the commuting O-D matrix.

4.1. Testing scenario and data preprocessing

The test scenario, illustrated in Section 1 of Fig. 1, is set within the mainland area of Manhattan. The selection of this location was driven by the accessibility of comprehensive historical trip datasets (NYC, 2016a, 2016b) from New York City and the area's rich concentration of commuting-related POIs. For our study, map data encompassing the road network and POIs were sourced from OpenStreetMap (OpenStreetMap contributors, 2017).

Following the initial data collection, we conducted meticulous corrections in SUMO (Lopez et al., 2018) for each road segment. This included adjustments to the number of lanes, connections between segments, and the geometries of intersections. To ensure accuracy, these corrections were guided by satellite imagery and Street View features from Google Earth Pro. The refined road network, as showcased in Fig. 1 (1b), comprises a total of 6,097 intersections and 12,296 road segments.

Out of a total of 33,793 POIs in the area, we identified 824 that were specifically relevant to commuting, as depicted in section (a) of Fig. 3. The selection of these POIs was based on their types (p_i^t), focusing predominantly on those associated with work and residential areas, such as banks, schools, universities, and offices. The specific types and quantities of these commuting-related POIs are detailed in section (b) of Fig. 3.

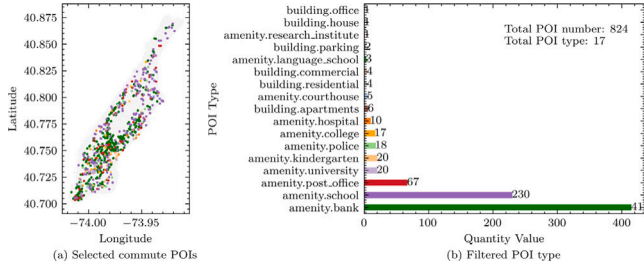


Fig. 3. Commuting POIs used in this paper. (a) Distribution of POIs by type. (b) Frequency distribution of POI types.

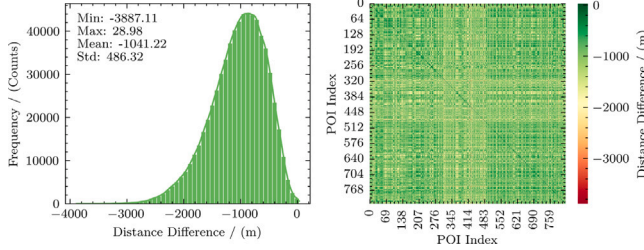


Fig. 4. Distribution of differences between Euclidean and path distances among POIs. (a) Distribution of distance differences (b) Heatmap of distance differences.

To define the commuting regions, the location of POIs were aligned with the geometry of road segments, thereby infusing the road network with commuting-specific characteristics. This was achieved by determining the proximity between the geographic coordinates of POIs (p_i^{geo}) and road segment geometry (r_m^{geo}), thus identifying the nearest road segment for each POI. This technique reflects how vehicles reach POIs along specific road segments in reality.

Subsequently, the Euclidean distance matrix (D^e) between POIs was calculated using Eq. (1). Meanwhile, the path distance matrix (D^p) among POIs was generated using the shortest path search functionality within sumolib. The difference between these two matrices is shown in Fig. 4.

For the analysis of O-D trips, as depicted in Section 2 of Fig. 1, we relied on the publicly available 2016 Yellow (NYC, 2016b) and Green Taxi dataset (NYC, 2016a) from New York City. This dataset is particularly valuable as it includes detailed coordinates for both pick-up and drop-off locations, which helps to accurately align the taxi trips with the commuting regions established in this study.

The processing of this dataset was executed in three stages. The first stage involved spatial filtering to isolate trips confined within the test scenario's geographical boundaries. The second stage concentrated on the dataset's temporal dimensions, extracting records of trips that occurred during the weekday morning peak hours in Manhattan, specifically between 7 and 10 AM. The final stage entailed a thorough cleansing of the dataset to remove any entries with null values. This comprehensive process results in approximately 15.6 million eligible O-D trips, as depicted in Fig. 5.

To determine valid routes for the O-D trips within the test scenario, the pick-up and drop-off points of each trip with corresponding road segments were aligned. This alignment was based on their respective geometries, utilising the principle of nearest Euclidean distance for precise matching. Following this, functionalities available in sumolib was used to find the optimal routes, leveraging the established relationships between the trip points and the road segments.

Regarding meteorological data, we use historical data from 2016, sourced from the Weather Underground website (WU, 2016). For Manhattan in 2016, there was only one observation point available for time-specific weather data. Given the relatively compact size of Manhattan, it is assumed that there is minimal variation in weather conditions

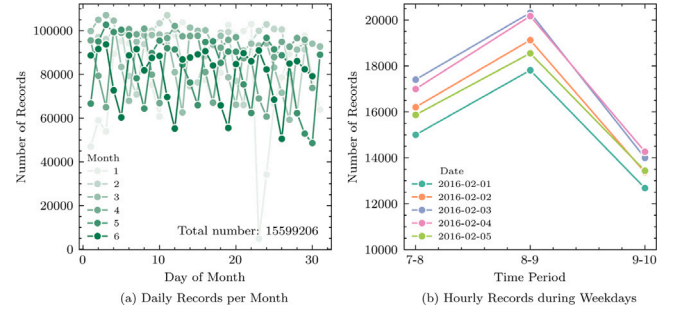


Fig. 5. Filtered O-D trips in the scenario. (a) Daily records per month. (b) Hourly records during weekdays (case study).

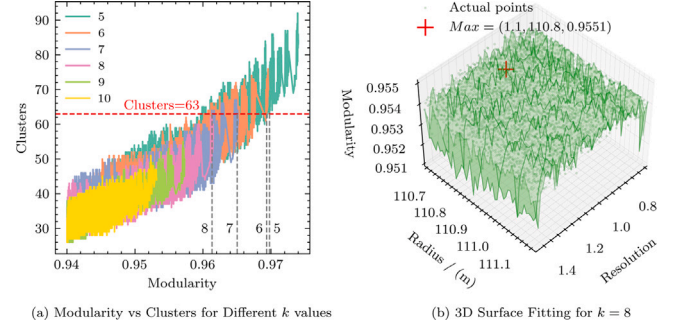


Fig. 6. Modularity results for different POI clustering parameters. (a) Modularity vs clusters for different k values. (b) 3D surface fitting for $k = 8$.

across different areas within the city. Therefore, we use the data from this single observation point to represent the weather conditions for all O-D regions in Manhattan. Notably, since 2018, multiple observation stations have been operational in Manhattan, suggesting that different O-D regions could have their distinct weather data in future.

4.2. Commute region division

In this section, we implement the methodologies detailed earlier to partition commute-related POIs and the road network within our designated test scenario. The subsequent discussion focusses on two primary aspects. First is the selection and justification of various parameters in use, and second is a comparative analysis of the proposed approach against other existing methods.

4.2.1. Commuting POI partition

The partitioning of commuting POIs as per Algorithm 1 primarily involves parameters such as graph construction requirements ($k, radius, n$) and edge weight coefficients ($\alpha_e, \alpha_s, \alpha_n, \alpha_j$), along with the Louvain clustering resolution r_{poi} . After extensive testing and analysis, we set the weight coefficients to $(\alpha_e, \alpha_s, \alpha_n, \alpha_j) = (0.4, 0.3, 0.2, 0.1)$. This denotes that $w_e^{poi} > w_s^{poi} > w_n^{poi} > w_j^{poi}$ in the composite edge weight (w^{poi}). We also set $n = 3$, which indicates that when the number of road segments between two POIs is less than 3, the edge weight should be greater.

Other parameters were optimised through testing, guided by modularity (defined as Eq. (4)). We evaluated modularity for different values of key parameters: the nearest neighbour count $k \in [5, 10]$ (step: 1); the clustering resolution $r_{poi} \in [0.7, 1.5]$ (step: 0.01); the Euclidean distance radius $radius \in [110.7, 111.2]$ (step: 0.01). This is because we found that the shortest distance between two POIs in communities was approximately 110.9 m in the test scenario.

The testing results, as depicted in Fig. 6(a), indicate that clustering with $k \in \{5, 6, 7, 8\}$ successfully achieves the classification target of 63 taxi zones in New York City. To strike a balance between accuracy and computational efficiency, we selected $k = 8$ for further

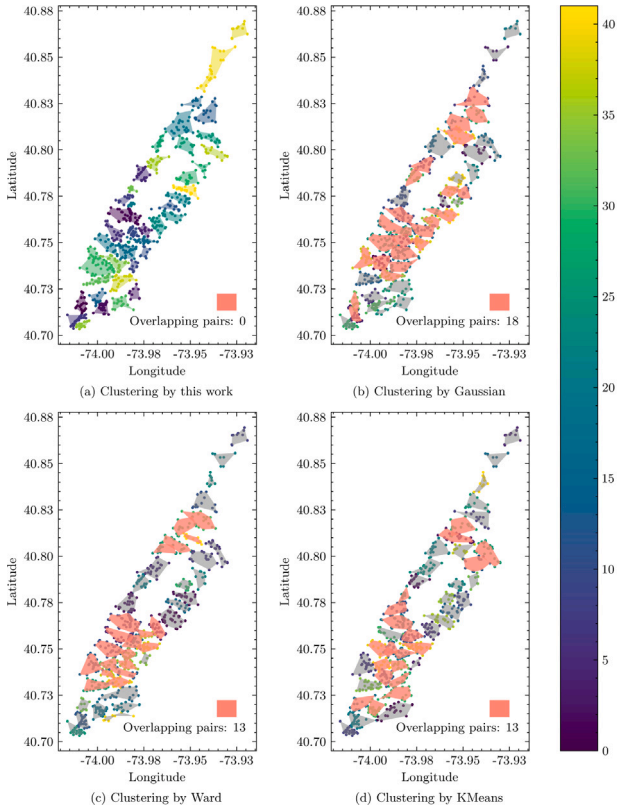


Fig. 7. The result of commuting POI clustering and comparison with results from other clustering methods.

Table 2
POI partition performance comparing with other methods.

Method	Modularity	Silhouette	Overlapping pairs
Louvain	0.938	0.199	1
MiniBatch	–	0.223	17
Spectral	–	0.229	14
Gaussian	–	0.236	18
Ward	–	0.241	13
K-Means	–	0.247	13
This work	0.954	0.234	0

analysis. Following this, we applied a 3D Gaussian Process fitting to $(r_{poi}, radius, modularity)$ to model the nonlinear relationships among these parameters. This method identified the optimal modularity value as 0.9551, which corresponded to $r_{poi} = 1.1$ and $radius = 110.8$, as shown in Fig. 6(b).

Finally, we used $(k, radius, n, r_{poi}) = (8, 110.8, 3, 1.1)$ and weight coefficients $(\alpha_e, \alpha_s, \alpha_n, \alpha_j) = (0.4, 0.3, 0.2, 0.1)$. This leads to a partition of POIs into 42 class, with the Visualisation presented in Fig. 7(a).

We also compared our method with other clustering approaches, including Louvain implemented in Orange3 (Demšar et al., 2013), Mini Batch K-Means (Sculley, 2010), Spectral Clustering (Ng et al., 2001), Gaussian Mixture (Reynolds, 2009), Ward (Ward, 1963), and K-Means (MacQueen, 1967). To evaluate these methods, we employed modularity as the principle metric. Additionally, the silhouette coefficient and the number of overlapping pairs were used as performance indicators. The outcomes of these comparative evaluations are detailed in Table 2.

In terms of modularity, the proposed method shows a notable superiority over the Louvain approach, indicating a more effective community detection. However, when assessed using the silhouette coefficient, this method performs worse than the Gaussian, Ward, or K-Means methods. It is important to note that in evaluations using

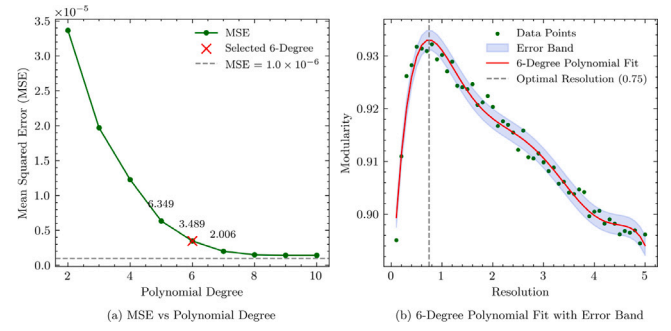


Fig. 8. Modularity for different resolution. (a) MSE results of polynomial fitting. (b) Fitting result using a 6th-degree polynomial.

concave functions for assessing POI clustering, the proposed method demonstrated a lower degree of overlap compared to these other methods. This aspect of our analysis, highlighting the relative strengths and weaknesses of the proposed approach compared to others, is visually represented in Fig. 7.

4.2.2. Road segment partition

The partition of the road segment as described in Algorithm 1 involves edge weight coefficients for graph construction, and the resolution (r_{edge}) for the Leiden algorithm. An optimal set of these coefficients are identified as $(\beta_b, \beta_p, \beta_s, \beta_n, \beta_d) = (0.01, 0.699, 0.1, 0.1, 0.1)$. This guarantees that connected road segments are appropriately linked by an edge. It also maximises the weights associated with POI clusters, while simultaneously considering the speed limits, lengths, and number of lanes of the road segments they comprise.

For the resolution r_{edge} in the Leiden algorithm, we used Eq. (8) to compute the modularity for $r_{edge} \in [0.1, 5.0]$ (step: 0.1). A polynomial fitting was applied to model the nonlinear relationship between r_{edge} and modularity.

Polynomial fittings ranging from the 2nd to the 10th degree were evaluated, employing MSE as the primary evaluation metric. The findings, as depicted in Fig. 8(a), revealed that there was no substantial improvement in MSE for polynomials beyond the 6th degree. Based on this observation, a 6th-degree polynomial fitting was selected, a choice that is demonstrated in Fig. 8(b). This particular fitting was instrumental in pinpointing the optimal value of $r_{edge} = 0.75$ for maximising modularity.

Finally, the partitioning process of road segments was executed according to Algorithm 1, incorporating the determined weight coefficients and setting $r_{edge} = 0.75$. The outcomes of this partitioning process are illustrated in Fig. 9. It is evident that the partitioned road segments align well with the commuting POI clusters. This correlation indicates both the similarities in road attributes and the underlying topological structure of the road network.

To evaluate the effectiveness of the proposed method, a comparative analysis with several community detection algorithms was conducted. This comparison focused on two key metrics: modularity and the number of generated clusters. These algorithms included Infomap (Rosvall & Bergstrom, 2008), Label Propagation (Raghavan et al., 2007), Spin-glass (Reichardt & Bornholdt, 2006), Leading Eigenvector (Newman, 2006), and Louvain (Blondel et al., 2008).

The results are presented in Table 3. They indicate that the proposed method closely aligns with the 42 identified POI clusters and attains the highest modularity value. This underscores its superior ability in accurately detecting and clustering community structures within the road network.

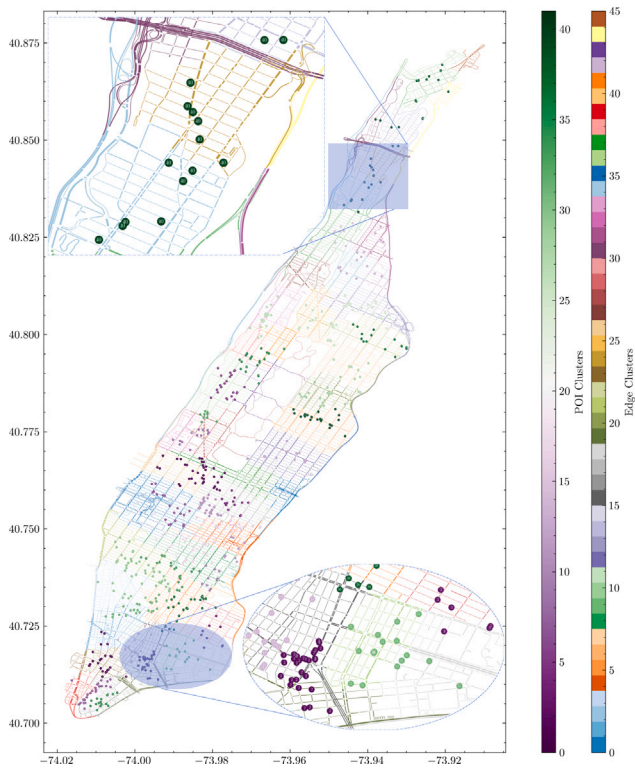


Fig. 9. The result of road segment partition.

Table 3
Road segment partitioning performance comparing with baselines.

Method	Modularity	Clusters
Infomap	0.449	1321
Label Propagation	0.636	1668
Spin-glass	0.848	25
Leading Eigenvector	0.782	47
Louvain	0.923	47
Proposed method	0.933	46

4.3. Commuting O-D matrix prediction

4.3.1. Experimental settings

Based on the clustering results and extracted trip information, the obtained dataset has a dimension of $543 \times 46 \times 8$ after the aforementioned data cleaning and pre-processing. The geometric information of each cluster was also incorporated to establish spatial connectivity. This dataset was divided into three subsets for training, validation and test purposes, with a ratio of 7:1.5:1.5.

The model presented was developed using the PyTorch framework. Its training employed the Adam optimiser and extended over 500 epochs. An initial learning rate of 0.002 was set, complemented by a decay parameter γ of 0.95, with a step size of 100. Data processing during each iteration was facilitated by a batch size of 16. Training and validation phases were executed on a workstation equipped with an AMD Ryzen Threadripper 3990X CPU, dual NVIDIA GeForce RTX 3090 GPUs, and 256 GB RAM. Where necessary, the LeakyReLU function was integrated as the activation function.

4.3.2. Benchmark models

In order to assess the performance of our proposed model, it was benchmarked against eight established models. This comparative analysis is designed to provide a comprehensive evaluation of the performance and efficiency of the proposed model, offering valuable insights into its advantages and contributions to the field. The baseline models selected for this comparison are summarised as follows.

- **ARIMA (Siarni-Namini et al., 2018)**: ARIMA represents a traditional method in time series analysis, highly regarded for its capability to model and predict using historical data. The primary advantages of ARIMA are its straightforwardness and effectiveness.
- **SVR (Bao et al., 2014)**: SVR, derived from the concepts of the Support Vector Machine, is adept at managing both linear and non-linear data relations. Its proficiency in addressing nonlinearities and focus on minimising margin errors render it a valuable asset in traffic forecasting applications.
- **LSTM (Tian & Pan, 2015)**: LSTM demonstrates a strong capability in integrating historical trends and correlations found within time series data. Such integration allows it to make robust predictions that are anchored in the temporal context, proving invaluable for accurate forecasting of traffic data.
- **STGCN (Yu et al., 2018)**: STGCN utilises graph theory to intricately map the connections among spatially distributed data points. It adeptly captures temporal patterns over time. The graph convolutional operations of STGCN facilitate the simultaneous capture of spatial and temporal features in traffic data, enhancing its efficacy in modelling intricate relationships and patterns within traffic flows.
- **ST-MAN (He et al., 2023)**: ST-MAN features a cross-modal attention mechanism for learning dynamic inter-modal correlations, a compact and efficient multimodal fusion framework to exploit both inter-modal and intra-modal correlations, and a refined spatio-temporal embedding mechanism that incorporates additional implicit information into the prediction process.
- **STSGCN (Song et al., 2020)**: STSGCN adeptly captures complex localised spatio-temporal correlations using a meticulously designed spatial-temporal synchronous modelling mechanism. The model incorporates multiple modules tailored for various time periods, efficiently addressing heterogeneity in localised spatio-temporal graphs.
- **4T-GCN (Li et al., 2024)**: 4T-GCN integrates four temporal CNN blocks and one GCN block, harnessing both temporal and spatial attributes from historical data for O-D demand forecasting. It incorporates time segment and day-of-the-week information as additional features.
- **AGCRN (Bai et al., 2020)**: AGCRN employs recurrent networks to capture spatio-temporal correlations through node adaptive parameter learning and data adaptive graph generation. It exhibits considerable enhancements in traffic flow prediction without depending on predefined graphs.

4.3.3. Performance evaluation

Table 4 presents a comparative analysis between the proposed model and eight benchmark models, categorising the results according to the prediction horizon. The proposed model outperforms other models in nearly all metrics and across various time frames, with the exceptions being slight deviations in sMAPE at longer intervals and a singular deviation in RMSE at the 3 h. This model achieves the lowest scores in the majority of evaluation metrics, highlighting its superior accuracy and reliability in predicting O-D data.

Among the existing models, AGCRN ranks as the second best in benchmark models, whereas SVR consistently demonstrates strong performance in sMAPE. However, the proposed model still exhibits an average improvement of 6.55% in MAE, 3.07% in RMSE, and a 4.06% reduction in sMAPE compared to AGCRN. A more pronounced improvement is observed when compared with ARIMA, which lags in both MAE and RMSE. Specifically, the proposed model shows a 29.93% reduction in MAE at the 1 h interval, escalating to 60.71% at the 4 h interval. A similar upward trend is seen in RMSE, with enhancements ranging from 45.07% to 67.22% as the prediction horizon extends.

These quantitative indicators vividly highlight the proposed model's superior precision and reliability in predicting O-D data. Its consistent superior performance over various intervals and metrics underscores its viability as an effective tool in transportation studies.

Table 4
O-D prediction performance comparison.

Model	1 h			2 h			3 h			4 h		
	MAE	RMSE	sMAPE	MAE	RMSE	sMAPE	MAE	RMSE	sMAPE	MAE	RMSE	sMAPE
ARIMA	63.21	140.15	10.29	93.02	207.88	13.04	109.23	243.20	14.35	148.86	324.00	17.39
SVR	61.83	99.04	10.39	79.14	143.48	10.96	84.48	152.59	11.30	75.51	137.76	10.96
LSTM	63.39	124.80	20.28	72.27	145.61	16.83	71.15	144.47	16.21	69.76	141.23	17.06
STGCN	59.70	104.51	16.23	74.92	135.89	17.56	76.24	142.98	17.46	83.58	147.16	17.64
ST-MAN	99.71	208.83	11.48	100.78	209.34	11.63	97.24	207.71	11.20	95.99	206.61	11.23
STSGCN	56.42	91.06	10.99	64.33	113.43	11.48	62.87	113.13	10.14	63.73	113.73	15.21
4T-GCN	50.36	89.35	20.31	61.28	112.21	21.27	60.69	117.79	18.96	59.54	116.32	16.93
AGCRN	50.96	83.78	10.56	54.73	94.35	9.76	54.91	93.86	10.50	62.08	111.08	10.49
Proposed model	44.27	77.03	9.87	51.23	91.07	10.73	54.46	97.03	11.24	58.14	106.18	11.22

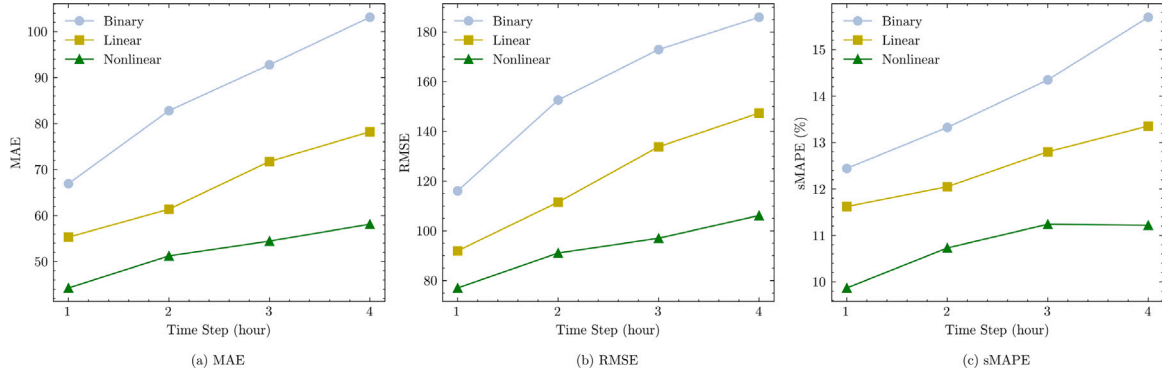


Fig. 10. Prediction performance comparison of incidence matrix.

4.3.4. Ablation study of weighted incidence matrix

The proposed weighted incidence matrix incorporates edge-dependent vertex weights, signifying varying degrees of vertex membership within a hyperedge. This feature is crucial for conveying vital network and route information, enabling the model to accurately learn spatial dependencies. To assess effectiveness, we conducted comparisons among three distinct types of incidence matrices. The first is an unweighted matrix, employing a binary representation to denote the relationship between vertices and a hyperedge, where 1 indicating inclusion of a vertex in a hyperedge.

Conversely, the remaining two methods calculate edge-dependent vertex weights based on the distance between vertices and the frequency of trips traversing each vertex. They employ both linear and nonlinear functions equipped with learnable coefficients. This methodology in constructing the weighted incidence matrix offers a thorough evaluation of the model's effectiveness across various configurations. It aids in identifying the most optimal representation for efficiently learning spatial dependencies. The nonlinear function is presented in Eq. (12), and the linear function can be described as,

$$h(v, e) = \alpha * \hat{f}_v + \beta * \hat{d}_e \quad (23)$$

where α and β are learnable parameters.

Results acquired from the ablation study are presented in Fig. 10. It unequivocally demonstrate that utilising a nonlinear edge-dependent vertex weight for the incidence matrix substantially elevates the performance of the model across all evaluated metrics. Specifically, in the context of MAE, the nonlinear weight function secures a 33.90% enhancement at the 1 h forecast interval and a 43.62% enhancement at the 4 h interval relative to the binary weight. This progression indicates that the benefit of the nonlinear representation amplifies as the prediction horizon widens, presumably attributed to its refined capability in delineating the complexities inherent to transportation networks as time advances. Against the backdrop of the linear matrix, the nonlinear methodology sustains a steady advantage, registering average improvements between 19.98% and 25.69% throughout the various time intervals, thereby highlighting the significance of nonlinear transformations in augmenting the precision of O-D predictions.

This pattern of improvement is mirrored in the additional evaluation metrics. For RMSE, the nonlinear incidence matrix achieves an average improvement of 40.19% over the binary weight and 22.50% over the linear weight across the four forecasted intervals. With respect to sMAPE, the improvements are 22.63% and 13.81% when compared to binary and linear weights, respectively.

The uniform superiority of the nonlinear incidence matrix, spanning all metrics and time intervals, reinforces its efficacy in capturing the nuanced, temporally evolving characteristics of transportation networks. The insights gleaned from this ablation study lend robust support to the integration of an edge-dependent vertex weight into the incidence matrix structure of hypergraphs, affirming its merit as a methodological enhancement in traffic data modelling.

4.3.5. Ablation study of contextual variables

As three additional contextual variables were incorporated to facilitate the prediction of commuting O-D matrix, an ablation study was undertaken to assess their impact on predictive accuracy. Each variable was added individually, while maintaining all other parameters constant. A baseline model, devoid of contextual variables, was also established.

Fig. 11 demonstrates that the inclusion of these contextual variables contributes positively to both MAE and RMSE metrics. In terms of MAE measurement, the day of the week yields a reduction ranging from 3.82% to 10.43% across various time steps, significantly surpassing the impact of hour (5.37% to 4.23%) and weather (4.94% to 5.25%). Meanwhile, for RMSE measurement, all variables show the greatest improvement at the 1 h mark, exceeding a 10% improvement over the baseline model. These enhancements gradually diminish to 5.78% for the day of the week, 7.4% for hour, and 2.38% for weather. However, the model integrating all features consistently outperforms in these two metrics, achieving improvements exceeding 10% at all time intervals. The evaluation of sMAPE presents variable results. Incorporating the day of the week or weather conditions results in enhanced performance, though the improvement is modest. In general terms, the day of the week and hour appear to exert a more substantial influence on prediction performance compared to weather. This discrepancy could be

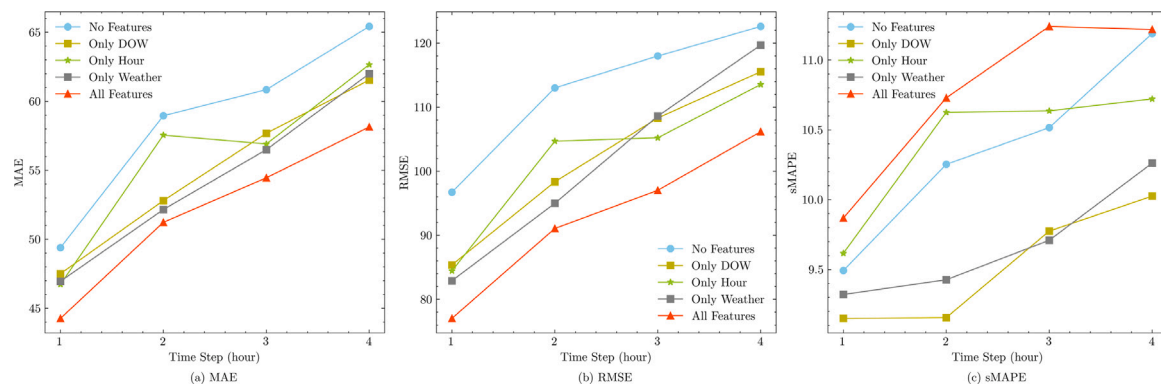


Fig. 11. Prediction performance comparison of contextual features.

attributed to the relatively infrequent occurrence of severe weather conditions in the collected dataset.

5. Conclusion

In this paper, we developed a novel approach integrating community detection with hypergraph representation learning to address the issue of estimating urban peak-hour commuting traffic demand. Specifically, we utilised the Louvain algorithm to group commute-related POIs according to their spatial location, and the Leiden algorithm on road networks with these POI clusters to partition commute O-D regions. Subsequently, we proposed a spatio-temporal dynamic weighted hypergraph model to predict peak-hour commuting O-D matrices, using the temporal and spatial characteristics from commute partitions, observed O-D trips, and meteorological data.

The approach has been evaluated employing historical O-D trip data, commute-related POI data, and road network data from Manhattan. Two significant outcomes emerged: firstly, the partitioning of all city road segments, which proved to be simpler and more comprehensive compared to traditional taxi zones. Secondly, a roughly 5% improvement in prediction accuracy over the state-of-the-art models, demonstrating that our model can more effectively capture the temporal and spatial characteristics of diverse heterogeneous data. Importantly, the success of this method suggests its potential capability to similar tasks involving temporal and spatial predictions with heterogeneous data, offering new avenues for urban planning and traffic management strategies.

Nevertheless, it is crucial to acknowledge the limitations of this study. Firstly, the testing scenario was confined to Manhattan, representing a single urban context. To further evaluate the effectiveness of our proposed method, future work will involve testing in diverse urban settings and datasets. For instance, we plan to conduct evaluations in cities like Shanghai, China, utilising datasets from taxi services and DiDi trips. Secondly, there is an opportunity to further enhance the precision of our prediction model. We plan to refine the existing model by incorporating attention mechanisms. This improvement is designed to mitigate the adverse impacts of non-recurrent variations in observed O-D trips, potentially caused by traffic incidents or major events. These forthcoming enhancements will not only overcome the present constraints but also substantially broaden the scope and utility of our study in the context of urban traffic forecasting.

CRedit authorship contribution statement

Yuhuan Li: Conceptualization, Methodology, Data Visualization, Writing – original draft. **Shaowu Cheng:** Supervision. **Yuxiang Feng:** Methodology, Data curation, Writing – original draft. **Yaping Zhang:** Supervision. **Panagiotis Angeloudis:** Supervision, Writing – review & editing. **Mohammed Quddus:** Supervision, Writing – review & editing. **Washington Yotto Ochieng:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by Sichuan Province Science and Technology Achievement Transfer Project 24ZHFS0115.

References

- Antelmi, A., Cordasco, G., Polato, M., Scarano, V., Spagnuolo, C., & Yang, D. (2024). A survey on hypergraph representation learning. *ACM Computing Surveys*, 56(1), 1–38. <http://dx.doi.org/10.1145/3605776>.
- Ashok, K., & Ben-Akiva, M. E. (2002). Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. *Transportation Science*, 36(2), 184–198. <http://dx.doi.org/10.1287/trsc.36.2.184.563>.
- Bai, L., Yao, L., Li, C., Wang, X., & Wang, C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in neural information processing systems: vol. 33*, (pp. 17804–17815).
- Bao, Y., Xiong, T., & Hu, Z. (2014). Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing*, 129, 482–493. <http://dx.doi.org/10.1016/j.neucom.2013.09.010>.
- Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. <http://dx.doi.org/10.1088/1742-5468/2008/10/P10008>.
- Chen, C. Y. T., Sun, E. W., Chang, M. F., & Lin, Y. B. (2023). Enhancing travel time prediction with deep learning on chronological and retrospective time order information of big traffic data. *Annals of Operations Research*, <http://dx.doi.org/10.1007/s10479-023-05223-7>.
- Chitra, U., & Raphael, B. (2019). Random walks on hypergraphs with edge-dependent vertex weights. Vol. 97, In *Proceedings of the 36th international conference on machine learning* (pp. 1172–1181). URL <https://proceedings.mlr.press/v97/chitra19a.html>.
- Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinović, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., & Zupan, B. (2013). Orange: Data mining toolbox in Python. *Journal of Machine Learning Research*, 14, 2349–2353.
- Dijkstra, E. W. (2022). A note on two problems in connexion with graphs. In *Edsger wybe Dijkstra: his life, work, and legacy* (pp. 287–290). <http://dx.doi.org/10.1145/3544585.3544600>.
- Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2021). A deep-learning model for urban traffic flow prediction with traffic events mined from Twitter. *World Wide Web*, 24(4), 1345–1368. <http://dx.doi.org/10.1007/s11280-020-00800-3>.
- Feng, Y., You, H., Zhang, Z., Ji, R., & Gao, Y. (2019). Hypergraph neural networks. Vol. 33, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 3558–3565). URL <https://doi.org/10.48550/arXiv.1809.09401>.
- Gao, F., Tang, J., & Li, Z. (2022). Effects of spatial units and travel modes on urban commuting demand modeling. *Transportation*, 49(6), 1549–1575. <http://dx.doi.org/10.1007/s11116-021-10219-y>.

- Ghosh, S., Halappanavar, M., Tumeo, A., Kalyanaraman, A., Lu, H., Chavarría-Miranda, D., Khan, A., & Gebremedhin, A. (2018). Distributed louvain algorithm for graph community detection. In *2018 IEEE international parallel and distributed processing symposium* (pp. 885–895). <http://dx.doi.org/10.1109/IPDPS.2018.00098>.
- Guo, Q., Li, L., & Ban, X. (2019). Urban traffic signal control with connected and automated vehicles: A survey. *Transportation Research Part C: Emerging Technologies*, 101, 313–334. <http://dx.doi.org/10.1016/j.trc.2019.01.026>.
- He, R., Li, L., Hua, B., Tong, J., & Tan, C. (2023). ST-MAN: Spatio-temporal multimodal attention network for traffic prediction. In *Knowledge science, engineering and management* (pp. 137–152). Switzerland: Springer Nature, 10.1007/978-3-031-40286-9_12.
- He, D., You, X., Feng, Z., Jin, D., Yang, X., & Zhang, W. (2018). A network-specific Markov random field approach to community detection. In *Proceedings of the thirty-second AAAI conference on artificial intelligence* (pp. 306–313). <http://dx.doi.org/10.1609/aaai.v32i1.11281>.
- Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2), 109–137. [http://dx.doi.org/10.1016/0378-8733\(83\)90021-7](http://dx.doi.org/10.1016/0378-8733(83)90021-7).
- Jiang, J., Yang, Y., Li, Y., Wang, R., & Zeng, S. (2022). Lane-level vehicle counting based on V2X and centimeter-level positioning at urban intersections. *International Journal of Intelligent Transportation Systems Research*, 20(1), 11–28. <http://dx.doi.org/10.1007/s13177-021-00271-4>.
- Jin, D., Li, B., Jiao, P., He, D., Shan, H., & Zhang, W. (2020). Modeling with node popularities for autonomous overlapping community detection. *ACM Transactions on Intelligent Systems and Technology*, 11(3), 1–23. <http://dx.doi.org/10.1145/3373760>.
- Kang, X., Li, X., Yao, H., Li, D., Jiang, B., Peng, X., Wu, T., Qi, S., & Dong, L. (2022). Dynamic hypergraph neural networks based on key hyperedges. *Information Sciences*, 616, 37–51. <http://dx.doi.org/10.1016/j.ins.2022.10.006>.
- Ke, J., Qin, X., Yang, H., Zheng, Z., Zhu, Z., & Ye, J. (2021). Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network. *Transportation Research Part C: Emerging Technologies*, 122, Article 102858. <http://dx.doi.org/10.1016/j.trc.2020.102858>.
- Li, Y., Zhang, Y., Feng, Y., Cheng, S., Li, L., & Ochieng, W. (2024). Short-term spatio-temporal forecasting of commuting origin-destination demand during peak hours. In *Transportation Research Board 103rd Annual Meeting*.
- Liang, J., Tang, J., Gao, F., Wang, Z., & Huang, H. (2023). On region-level travel demand forecasting using multi-task adaptive graph attention network. *Information Sciences*, 622, 161–177. <http://dx.doi.org/10.1016/j.ins.2022.11.138>.
- Liu, Z., Liu, Z., & Fu, X. (2022). Dynamic origin-destination flow prediction using spatial-temporal graph convolution network with mobile phone data. *IEEE Intelligent Transportation Systems Magazine*, 14(5), 147–161. <http://dx.doi.org/10.1109/MTS.2021.3082397>.
- Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y. P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wiessner, E. (2018). Microscopic traffic simulation using SUMO. In *2018 21st international conference on intelligent transportation systems* (pp. 2575–2582). <http://dx.doi.org/10.1109/ITSC.2018.8569938>.
- Ma, Y., Zhong, J., & Zhu, N. (2023). Weighted hypergraph learning and adaptive inductive matrix completion for SARS-CoV-2 drug repositioning. *Methods*, 219, 102–110. <http://dx.doi.org/10.1016/j.jymeth.2023.10.002>.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (pp. 281–297).
- Newman, M. E. J. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), Article 036104, URL <https://doi.org/10.1103/PhysRevE.74.036104>.
- Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2), Article 026113. <http://dx.doi.org/10.1103/PhysRevE.69.026113>.
- Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*: Vol. 14, (pp. 849–856).
- Nowozin, S. (2010). Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3), 185–365. <http://dx.doi.org/10.1561/06000000033>.
- NYC (2016a). 2016 Green taxi trip data. <https://data.cityofnewyork.us/Transportation/2016-Green-Taxi-Trip-Data/hvrb-b6nb>.
- NYC (2016b). 2016 Yellow taxi trip data. <https://data.cityofnewyork.us/Transportation/2016-Yellow-Taxi-Trip-Data/k67s-dv2t>.
- OpenStreetMap contributors (2017). Planet dump retrieved from <https://planet.osm.org>. URL <https://www.openstreetmap.org>.
- Pamula, T., & Zochowska, R. (2023). Estimation and prediction of the OD matrix in uncongested urban road network based on traffic flows using deep learning. *Engineering Applications of Artificial Intelligence*, 117, Article 105550. <http://dx.doi.org/10.1016/j.engappai.2022.105550>.
- Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76, Article 036106. <http://dx.doi.org/10.1103/PhysRevE.76.036106>.
- Reichardt, J., & Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74(1), Article 016110, URL <https://doi.org/10.1103/PhysRevE.74.016110>.
- Reynolds, D. A. (2009). Gaussian mixture models. Vol. 741, In *Encyclopedia of biometrics* (pp. 659–663). http://dx.doi.org/10.1007/978-0-387-73003-5_196.
- Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4), 1118–1123, URL <https://doi.org/10.1073/pnas.0706851105>.
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65. [http://dx.doi.org/10.1016/0377-0427\(87\)90125-7](http://dx.doi.org/10.1016/0377-0427(87)90125-7).
- Sculley, D. (2010). Web-scale K-means clustering. In *Proceedings of the 19th international conference on world wide web* (pp. 1177–1178). <http://dx.doi.org/10.1145/1772690.1772862>.
- Siarni-Namini, S., Tavakoli, N., & Siarni Namin, A. (2018). A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications* (pp. 1394–1401). <http://dx.doi.org/10.1109/ICMLA.2018.00227>.
- Soares, E. F. D. S., Revoredo, K., Baião, F., Quintella, C. A. d. M. S., & Campos, C. A. V. (2019). A combined solution for real-time travel mode detection and trip purpose prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(12), 4655–4664, URL <https://doi.org/10.1109/ITS.2019.2905601>.
- Song, C., Lin, Y., Guo, S., & Wan, H. (2020). Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. Vol. 34, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 914–921). <http://dx.doi.org/10.1609/aaai.v34i01.5438>.
- Sun, B., Ermagun, A., & Dan, B. (2017). Built environmental impacts on commuting mode choice and distance: Evidence from Shanghai. *Transportation Research Part D: Transport and Environment*, 52, 441–453. <http://dx.doi.org/10.1016/j.trd.2016.06.001>.
- Tian, Y., & Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE international conference on smart city/socialCom/ sustainCom* (pp. 153–158). <http://dx.doi.org/10.1109/SmartCity.2015.63>.
- Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1), 5233. <http://dx.doi.org/10.1038/s41598-019-41695-z>.
- Viñas, R., Joshi, C. K., Georgiev, D., Lin, P., Dumitrascu, B., Gamazon, E. R., & Liò, P. (2023). Hypergraph factorization for multi-tissue gene expression imputation. *Nature Machine Intelligence*, 5(7), 739–753. <http://dx.doi.org/10.1038/s42256-023-00684-8>.
- Wang, H., Zhang, R., Cheng, X., & Yang, L. (2022). Hierarchical traffic flow prediction based on spatial-temporal graph convolutional network. *IEEE Transactions on Intelligent Transportation Systems*, 23, 16137–16147. <http://dx.doi.org/10.1109/TITS.2022.3148105>.
- Wang, Y., & Zhu, D. (2024). A hypergraph-based hybrid graph convolutional network for intracity human activity intensity prediction and geographic relationship interpretation. *Information Fusion*, 104, Article 102149. <http://dx.doi.org/10.1016/j.inffus.2023.102149>.
- Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58, 236–244. <http://dx.doi.org/10.1080/01621459.1963.10500845>.
- WU (2016). 2016 New York weather history. URL <https://www.wunderground.com/history/daily/us/ny/new-york-city/KLGA/date/2016-1-1>.
- Yan, X., Song, T., Jiao, Y., He, J., Wang, J., Li, R., & Chu, W. (2023). Spatio-temporal hypergraph learning for next POI recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 403–412). <http://dx.doi.org/10.1145/3539618.3591770>.
- Ye, Q., Feng, Y., Candela, E., Escibano, J., Stettler, M., & Angeloudis, P. (2022). Spatial-temporal flows-adaptive street layout control using reinforcement learning. *Sustainability*, 1–22. <http://dx.doi.org/10.3390/su14010107>.
- Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence* (pp. 3634–3640). <http://dx.doi.org/10.24963/ijcai.2018/505>.
- Zeng, J., & Yu, H. (2018). A scalable distributed louvain algorithm for large-scale graph community detection. In *2018 IEEE international conference on cluster computing* (pp. 268–278). <http://dx.doi.org/10.1109/CLUSTER.2018.00044>.
- Zhang, D., & Xiao, F. (2023). Dynamic auto-structuring graph neural network: A joint learning framework for origin-destination demand prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(4), 3699–3711. <http://dx.doi.org/10.1109/TKDE.2021.3135898>.
- Zhang, D., Xiao, F., Shen, M., & Zhong, S. (2021). DNEAT: A novel dynamic node-edge attention network for origin-destination demand prediction. *Transportation Research Part C: Emerging Technologies*, 122, Article 102851. <http://dx.doi.org/10.1016/j.trc.2020.102851>.
- Zhu, Y., & Segarra, S. (2023). Hypergraphs with edge-dependent vertex weights: p-Laplacians and spectral clustering. *Frontiers in Big Data*, 6, <http://dx.doi.org/10.3389/fdata.2023.1020173>.
- Zou, X., Zhang, S., Zhang, C., Yu, J. J. Q., & Chung, E. (2022). Long-term origin-destination demand prediction with graph deep learning. *IEEE Transactions on Big Data*, 8(6), 1481–1495. <http://dx.doi.org/10.1109/TBDA.2021.3063553>.