Received: 4 December 2024 | Accepted: 2 March 2025

DOI: 10.1111/mice.13458

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

RESEARCH ARTICLE

Origin–destination prediction via knowledge-enhanced hybrid learning

Zeren Xing[1,2] | **Edward Chung**[2] | **Yiyang Wang**[1] | **Azusa Toriumi**[3] | **Takashi Oguchi**[3] | **Yuehui Wu**[4]

[1]Department of Civil Engineering, The University of Tokyo, Tokyo, Japan

[2]Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong, China

[3]Institute of Industrial Science, The University of Tokyo, Tokyo, Japan

[4]College of Transportation Engineering, Dalian Maritime University, Dalian, China

Correspondence
Edward Chung, Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong, China.
Email: edward.cs.chung@polyu.edu.hk

Funding information
JST SPRING, Grant/Award Number: JPMJSP2108; Key Technologies of Traffic Signal Control Evaluation Diagnosis-Optimization Driven by Artificial Intelligence and Digital Twin, Innovation and Technology Fund - Mainland-Hong Kong Joint Funding Scheme (ITF-MHKJFS), Grant/Award Number: MHP/038/23

Abstract

This paper proposes a novel origin–destination (OD) prediction (ODP) model, namely, knowledge-enhanced hybrid spatial–temporal graph neural networks (KE-H-GNN). KE-H-GNN integrates a deep learning predictive model with traffic engineering domain knowledge and a multi-linear regression (MLR) module for incorporating external factors. Leveraging insights from the gravity model, we propose two meaningful region partitioning strategies for reducing data dimension: election districts and K-means clustering. The aggregated OD matrices and graph inputs are processed using an long short-term memory network to capture temporal correlations and a multi-graph input graph convolutional network module to capture spatial correlations. The model also employs a global–local attention module, inspired by traffic flow theory, to capture nonlinear spatial features. Finally, an MLR module was designed to quantify the relationship between OD matrices and external factors. Experiments on real-world datasets from New York and Tokyo demonstrate that KE-H-GNN outperforms all the baseline models while maintaining interpretability. Additionally, the MLR module outperformed the concatenation method for integrating external factors, regarding both performance and transparency. Moreover, the election district-based partitioning approach proved more effective and simpler for practical applications. The proposed KE-H-GNN offers an effective and interpretable solution for ODP that can be practically applied in real-world scenarios.

1 | INTRODUCTION

With rapid urbanization, transportation systems are evolving and expanding to accommodate growing travel demands (D. Zhang et al., 2021). Accurate origin–destination (OD) prediction (ODP) is crucial for improving the efficiency and safety of urban transportation systems (Zou et al., 2022), as it provides a detailed understanding of real-time travel demand patterns. For instance, by predicting the number of passenger OD demands, taxi service platform operators can dynamically allocate resources, such as dispatching additional vehicles to high-demand regions. This ensures better service availability, minimizes vehicle idle times, reduces passenger waiting times, and thereby enhances overall system performance.

ODP has been an attractive research topic, and a considerable amount of research has been conducted over the past decades (Zhu et al., 2019). Unlike traffic flow predictions that only forecast the traffic counts at specific points (or areas), ODP is more challenging (Ke et al., 2021; D. Li et al., 2023) as it predicts the flow simultaneously from origins and destinations.

In the current literature, spatial–temporal (ST) deep learning models are the predominant approach for ODP due to their capability to model nonlinear relationships and capture spatiotemporal patterns in traffic data. Various methods have been developed, such as convolutional neural network (CNN)-based models (Chu et al., 2020; Liu et al., 2019) and graph convolutional network (GCN)-based models (B. Huang et al., 2023; Ke et al., 2021; Shi et al., 2020; Wang et al., 2019; Yao et al., 2021; Zou et al., 2022). While ST deep learning models are well-established in the literature, much of the existing research has primarily focused on enhancing prediction accuracy. Some studies have begun addressing real-world challenges, such as handling missing data (Qu et al., 2023, 2024), and transferring knowledge across cities (Lu et al., 2022; Yao et al., 2019).

Nonetheless, the field of ODP still faces significant challenges:

1. Data sparseness and region partition: Compared to traffic regional demand prediction, ODP presents a major challenge due to data sparseness, with many zero values (Peng et al., 2023). The data sparseness issue needs to be addressed for two main reasons: First, those low-value ODs may not meaningfully impact traffic management since those dynamic traffic control strategies are mainly developed for high-demand regions. Second, including low-value ODs in the prediction process reduces the overall accuracy, as these pairs often add noise that affects the prediction of more significant ODs. By reducing the dimension of the OD matrix, the prediction model can learn more effectively from the most important information. Some existing work (Hu et al., 2020; Wang et al., 2019) applied mathematical approaches to address the data sparseness problem. However, these methods often increase computational complexity, especially when dealing with large-scale networks with high-dimensional OD data. Another commonly used approach to reduce the dimension of the OD matrix is through region partition. Existing literature typically divides the target city into uniform grids, such as squares (Liu et al., 2019; R. Zhang et al., 2022), or hexagons (Yang et al., 2021). The prediction tasks were based on those aggregated OD matrices. This region partition approach, while convenient, limits the ability to capture the administrative and functional attributes of the regions (Ke et al., 2021) and disregards the spatial heterogeneity across different parts of the city. Furthermore, the predicted grid-to-grid ODs are hard to correlate with real-world traffic patterns or network characteristics, making it difficult to apply and interpret for real planning tasks. Hence, how to effectively capture essential information from sparse OD data, with practical significance and implications, remains a challenge in the research community.
2. Inclusion of external factors: External factors such as weather, weekends, and holidays, play an important role in ODP. Existing research on ODP incorporating external factors (Tong et al., 2017; J. Xu et al., 2018; Zou et al., 2022) purely concatenate or fuse those external data with OD data. Nonetheless, this approach cannot well present the relationship between OD data and external factor data and makes it difficult to quantify the effects of external factors. Quantifying the effects of external factors, such as rain, is important from a planning and policy-making perspective. Therefore, how to effectively incorporate external factors and understand their contribution to the ODP remains an open question.
3. Model performance and interpretability: Deep-learning models have been proven effective for ODP tasks since they offer high prediction accuracy and the ability to model complex traffic patterns (H. Xu et al., 2024). Yet, they also present certain limitations due to their "black box" nature. While they can capture complex relationships within the traffic data, they offer limited insight into how predictions are made. For transportation planners, this drawback makes it difficult to validate the model or explain its outcomes to stakeholders. Consequently, it is of great practical significance to establish models that can achieve a balance between prediction performance and interpretability.

To address the previously mentioned challenges, this research proposed a novel ODP model, namely, knowledge-enhanced hybrid ST graph neural networks (KE-H-GNN), designed to balance prediction performance and interpretability. Unlike existing works that rely solely on deep learning models, KE-H-GNN combines a deep learning predictive model incorporating traffic engineering domain knowledge with a multi-linear regression (MLR) module.

Specifically, based on the knowledge derived from the gravity model, we first propose to use election district and K-means clustering as the bases for reducing the dimension of OD matrices, which addressed the data sparseness and the regional partitioning issues simultaneously. Regarding the deep learning predictive model, it is based on LSTM and GCN. We first applied an LSTM

network to capture the temporal correlations among OD matrices. Then a multi-graph input GCN module was developed to learn the spatial correlations, which consists of three branches: two branches of GCNs for static graphs and one for dynamic graphs using bi-directional GCN (BDGCN) layers. Finally, a global–local attention module was applied to comprehensively capture the spatial features among the traffic data. Meanwhile, an MLR approach was applied to model and quantify the relationship between the predicted OD matrices and external factors. To verify the effectiveness and robustness of the proposed framework, extensive experiments were conducted on two large-scale cities: New York and Tokyo. Furthermore, interpretation analyses were conducted to estimate the effects of external factors.

To the best of our knowledge, no prior research has attempted to use election districts and K-means clustering to reduce the dimension of OD matrices. Unlike prior works that use grid-based region partitioning (e.g., Liu et al., 2019; R. Zhang et al., 2022; Zou et al., 2022), the utilization of election districts and K-means clustering as the aggregation bases for region partitioning is more meaningful. Furthermore, limited research incorporated traffic flow theory into the ODP model. X. Huang et al. (2023) first proposed a local flow attention module based on vehicle conservation law. They applied convolutions and used convolutional kernels to represent neighborhood ranges, which is not applicable for graph-based data. We have developed a novel local attention module based on the vehicle conservation law, called vehicle conservation attention (VCA). Compared to Huang et al. (2023), our proposed approach directly uses an adjacency matrix to reflect neighboring relationships, which is suitable to model graph-based data. Finally, no work has been done to quantify the relationship between traffic data and external factors using an MLR approach.

In summary, the main contributions of this study are as follows:

1. We propose a novel ODP model by combining deep learning and MLR models while incorporating traffic engineering domain knowledge. The proposed model achieves a balance between prediction performance and interpretability.
2. We introduce traffic engineering domain knowledge to enhance deep learning algorithms, forming a comprehensive framework for ODP. Specifically, we propose a knowledge-driven approach for reducing data dimensions, with practical significance and implications. Additionally, we propose a novel global–local attention module following the traffic flow theory. This module is effective for extracting spatial features among traffic data.
3. We propose an MLR approach for including external factor data. This module can model the relationship between traffic data and external factors and quantify the effects of individual external factors, thus contributing to enhancing the interpretability of the deep learning model.

The rest of the paper is structured as follows: Section 2 provides a comprehensive review of the relevant literature on ODP. Section 3 describes the preliminaries. Section 4 introduces the proposed ODP framework. Section 5 presents the data, evaluation metrics, experiment settings, and baseline models, followed by the results and discussion in Section 6. Section 7 presents the conclusions and future works.

2 | LITERATURE REVIEW

ODP is crucial for effective urban planning and transportation management. Over the decades, various methods have been developed to forecast OD matrices, ranging from traditional models to advanced learning-based approaches. Traditionally, the four-step model has been widely used for ODP (McNally, 2000), serving as a foundational framework for understanding and estimating travel demand. Building on this foundation, this section provides an overview of learning-based approaches, including statistical approaches, machine learning approaches, and deep learning approaches, highlighting their advantages and limitations in the context of ODP.

One of the simplest and most classical OD forecasting methods is to predict future ODs from historical traffic characteristics, namely, the historical average (HA) model. Traditional statistical approaches to ODP are generally based on parametric methods, aiming to capture the temporal correlations within traffic data. For instance, Moreira-Matias et al. (2013) used Poisson and autoregressive integrated moving average models to forecast taxi demand based on Global Positioning System (GPS) streaming data. X. Li et al. (2017) proposed a hybrid taxi ODP algorithm (NMF-AR) by combining the nonnegative matrix factorization algorithm and the autoregressive model. The advantage of the above prediction models is that they are established based on sound mathematical principles, making them understandable and allowing people to interpret how predictions were made. However, it is difficult for statistical models to capture the complex relationships among traffic data, limiting their ability to achieve high prediction accuracy, especially in dynamic traffic environments.

Existing literature demonstrated that nonlinear machine learning models should be applied in the traffic

and ODP tasks (Yaguang Li et al., 2018; Vlahogianni et al., 2014), such as support vector machines (SVMs) and artificial neural networks (ANN). Y. Li et al. (2017) used a least squares SVM model to predict taxi traffic demand. Tong et al. (2017) proposed a simple unified linear regression model to predict the unit original taxi demand with high-dimension features for a large-scale online taxicab platform. Mukai & Yoden (2012) applied an ANN in predicting taxi demand through taxi probe data in the central areas of Tokyo. Compared with statistical models, machine learning approaches provide certain improvements in prediction accuracy. However, when dealing with large-scale datasets, these models consume large amounts of computing resources, thus affecting their practical applicability (Zou et al., 2023).

As computing power and data availability have increased, deep learning-based methods have become popular for OD estimation and forecasting. To better capture the nonlinear characteristics of traffic data, researchers have developed various types of deep learning-based models, such as recurrent neural networks (RNNs), and CNNs. For instance, Zhao et al. (2018) proposed a long short-term memory (LSTM) and a linear regression model and demonstrated the effectiveness of the proposed model against traditional ODP methods. J. Xu et al. (2018) proposed a sequence learning model combining an LSTM and a mixture density network with external factors. The proposed model predicts the probability distribution of taxi demands in all areas instead of directly predicting the taxi ODs. Liu et al. (2019) proposed a convolutional LSTM (ConvLSTM) based model, namely, contextualized spatial–temporal network for taxi OD demand prediction. The model simultaneously considers three components including local spatial context, temporal evolution context, and global correlation context. Chu et al. (2020) proposed a multi-scale ConvLSTM network to capture both temporal and spatial features from historical data and metadata. In addition, they used permutation and matricization to reduce the data dimension of the OD tensor.

CNN-based approaches can extract local spatial relationships but are not capable of modeling semantic connections between two zones (Ke et al., 2021). For example, two regions may be geographically far but can be semantically correlated if they serve similar functions (e.g., residential or commercial areas). Therefore, recent studies developed a GCN in the field of ODP. Wang et al. (2019) proposed an LSTM and GCN-based multi-task learning model, to predict the inflow and outflow simultaneously. They designed and grid-embedding framework and pre-weighted functions for its aggregating process to handle the data sparsity problem. Ke et al. (2021) proposed an ST encoder–decoder residual multi-GCN for predicting taxi ODs. They designed a residual multi-GCN combined with an LSTM network for encoding spatial correlations and temporal correlations. Then the Residual Multi-Graph Convolutional network (RMGC) were reapplied for decoding and prediction. Shi et al. (2020) developed a multi-perspective GCN (MPGCN). MPGCN consists of three branches, where each branch contains an LSTM and a BDGCN with different graph inputs to capture the complex ST traffic correlations. They included both static and dynamic graphs and used an average strategy to obtain the final predicted OD flow. Yao et al. (2021) introduced spatial interaction GCN (SIGCN), consisting of a spatial representation layer and an encoder–decoder framework. SIGCN employs geographical unit embedding and negative sampling, which contributed to enhancing the prediction accuracy. Zou et al. (2022) proposed a CNN and GCN-based model called ST graph deep learning (ST-GDL) to forecast OD matrices over both short and long-term horizons with external features. The proposed model can capture multiple temporal features including closeness, periodicity, and trend. A novel GCN block was introduced to extract dynamic spatial relationships within the network. Han et al. (2022) proposed a novel approach called continuous-time and multi-level dynamic graph representation learning (CMOD). It continuously updates traffic node states, avoiding the information loss associated with traditional fixed time windows. CMOD also adaptively captures spatial dependencies through a multi-level structure. R. Zhang et al. (2022) proposed a dynamic node representation learning model. A hierarchical memory updater was designed to handle both fine-grained and macro-level temporal dynamics. Also, a random walk-based OD node embedding module was proposed to differentiate semantic meanings between the origin nodes and destination nodes. B. Huang et al. (2023) introduced a BDGCN and transformer-based model (ODformer). Specifically, they designed a novel OD attention and period sparse self-attention to address the varying ST dependency problem for long sequence OD forecasting. Deep learning models have advanced performances for ODP tasks. However, their "black box" nature limits interpretability, posing challenges for practical applications. Therefore, it is crucial to establish models that balance predictive performance with interpretability, ensuring the outputs are both reliable and explainable for decision-making.

In summary, statistical methods offer interpretable frameworks but lack the capacity to model complex ST dependencies, leading to limited accuracy. Machine learning methods improve upon this by capturing nonlinear relationships, offering better prediction performance. However, they often fall short in dealing with multi-dimensional data, which limits their applicability in dynamic traffic scenarios. On the other hand, deep learning approaches have advanced predictive performance by capturing nonlinear ST correlations. Nonetheless, as

mentioned in the previous section, critical challenges remain in handling sparse OD matrices, quantifying the effects of external factors, and balancing prediction accuracy with interpretability. These gaps highlight the need for a developing novel hybrid approach, such as KE-H-GNN, to address these challenges effectively.

3 | PRELIMINARIES

3.1 | Problem statement

ODP problem: The ODP problem aims to predict future OD matrices based on observed data and external factors. Given:

1. Historical t observed OD matrices $\{M_1, M_2, \dots, M_t\}$ with dimension $t \times \mathbb{N} \times \mathbb{N} \times 1$. Here, t is the number of observed time steps, and $\mathbb{N}$ is the number of nodes or OD zones. In this research, the interval between consecutive time steps is 1 h.
2. External factor data $\{E_1, E_2, \dots, E_t\}$ with dimension $t \times \mathbb{N} \times \mathbb{N} \times \mathbb{P}$, where $\mathbb{P}$ is the number of external factors being considered.

The target is to predict the OD matrix $\{M_{t+1}, M_{t+2}, \dots, M_{t+\tau}\}$ in the next τ time steps. The problem is equivalent to learn a function $F(\cdot)$ such that:

$$F(M_1, M_2, \dots, M_t;\ E_1, E_2, \dots, E_t) \Rightarrow M_{t+1},\ M_{t+2}, \dots,\ M_{t+\tau} \tag{1}$$

3.2 | GCN and BDGCN

Traditional CNN approaches are restricted to regular grids, making them unsuitable for graph-structured data. Kipf and Welling (2016) introduced a simplified spectral GCN, which approximates spectral graph convolutions by restricting the Chebyshev polynomial filter to first-order neighbors. This significantly reduced computational complexity, making GCNs more efficient and applicable to graph-structured data tasks.

Considering a traffic network that is partitioned into $\mathbb{N}$ traffic zones, we define the target traffic network as an undirected graph $G = (V, E, A)$. V represents a finite set of nodes ($|V| = \mathbb{N}$) and each node corresponds to a traffic zone; E denotes a set of edges, which captures the connections between these nodes; $A \in R^{\mathbb{N}\times\mathbb{N}}$ represents the adjacency matrix of graph G.

In traditional spectral graph theory, a graph is characterized by its Laplacian matrix. By examining the symmetric normalized Laplacian matrix and its eigenvalues, the properties of the graph structure can be inferred. The Laplacian matrix is defined as

$$L = I_{\mathbb{N}} - D^{-1/2} A D^{-1/2} \tag{2}$$

$$D\ (i,i) = \sum_j A\,(i,j) \tag{3}$$

$I_{\mathbb{N}}$ is the identity matrix of dimension $\mathbb{N} \times \mathbb{N}$, D denotes the diagonal degree matrix of the adjacency matrix A. However, eigenvalue decomposition is computationally expensive when the dimension of the graph becomes large. To address this, Kipf and Welling (2016) approximated it using first-order Chebyshev polynomial expansion, which contributed to simplifying the problem. The core idea behind GCNs is the graph convolution operation, which is given as

$$X^{(l+1)} = f(X^{(l)},\ A) = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}X^{(l)}W^{(l)}\right) \tag{4}$$

$$\tilde{A} = A + I_{\mathbb{N}} \tag{5}$$

In the above equations, $X^{(l)}$ denotes the feature matrix of layer l. σ is the activation function (e.g., Rectified Linear Unit (ReLU)). Equation (5) allows nodes to incorporate their own features during the information propagation process. $W^{(l)}$ is a learnable weight matrix. Moreover, BDGCN is an extended form of GCN that can obtain two-dimensional correlations for both origins and destinations. The formulation is defined as follows:

$$X^{(l+1)} = \sigma\left(\sum_{i=0}^{K-1}\sum_{j=0}^{K-1} X^{(l)} \times T_i\,(L_O) \times T_j\,(L_D) \times W^{(l)}\right) \tag{6}$$

Here, $T_i(L)$ denotes the ith order Chebyshev polynomial. For each BDGCN layer, the neighbor information propagates across K hops between either origin or destination regions. Through the above process, GCNs and BDGCNs aggregate information from each node's neighbors and pass it through layers of transformations, thereby capturing the dependencies and interactions between nodes.

4 | METHODOLOGIES

4.1 | KE-H-GNN

Figure 1 presents the overall framework of the KE-H-GNN. KE-H-GNN combines a deep learning model with an MLR module. Regarding the deep learning predictive model, it is based on LSTM and GCN. We introduce three traffic engineering domain knowledge that is useful for ODP and incorporate them in the proposed KE-H-GNN model.

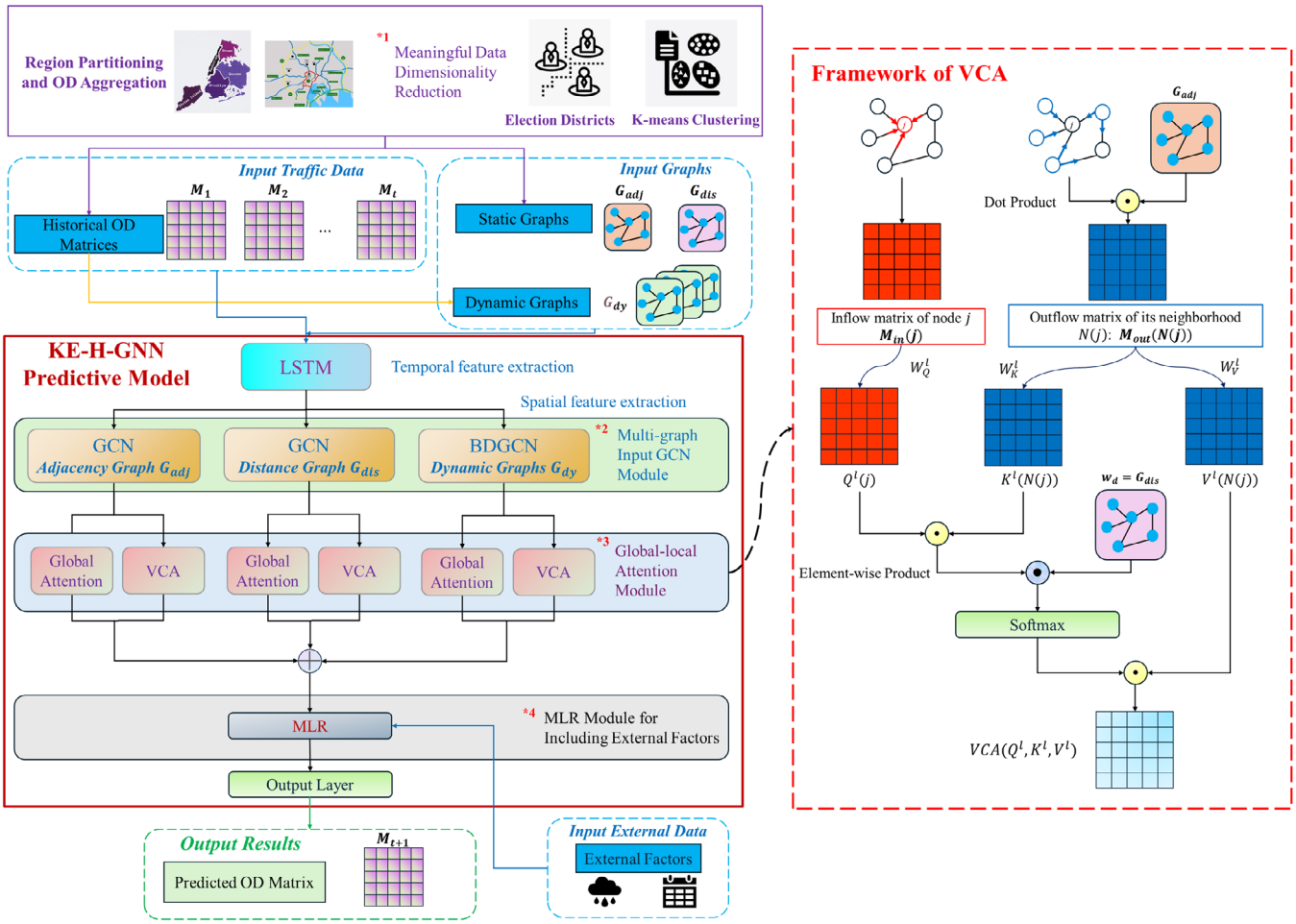


FIGURE 1 Framework of the knowledge-enhanced hybrid spatial–temporal graph neural networks (KE-H-GNN). BDGCN, bi-directional graph convolutional network; GCN, graph convolutional network; LSTM, long short-term memory; MLR, multi-linear regression; OD, origin–destination; VCA, vehicle conservation attention.

Details of each module are explained in the following subsections.

Before inputting into the model, we first partitioned the target networks and aggregated the historical OD matrices using election district and K-means clustering (Section 4.2). The aggregated historical OD matrices and static graphs were inputted into the prediction model. We first constructed dynamic graphs based on the correlations of the historical OD matrices. Then an LSTM network was applied to capture the temporal correlations among the historical OD matrices. Moreover, we designed a multi-graph input GCN module with three branches of different graph inputs to learn the spatial correlations (Section 4.3). Specifically, two branches of GCN layers were applied for static graphs; and a branch of BDGCN layers was applied for dynamic graphs. Moreover, a global–local attention module, consisting of global self-attention and VCA, was applied after each GCN branch to comprehensively capture spatial features among the nodes (Section 4.4). Finally, an MLR module was used to model and quantify the relationship between the predicted OD matrices and external factors (Section 4.5).

4.2 | Meaningful data dimensionality reduction

Knowledge 1: The flow between OD pairs depends on their masses (e.g., population, economic activity, etc.) and their distance according to the traditional theory-based ODP model, such as the gravity model.

The gravity model considers the following two important aspects of travel behavior:

1. The "mass" signifies the potential of an origin or destination to generate or attract trips. Larger masses indicate higher trip generation capability.
2. Distance acts as an obstacle for travelers. As distance increases, the likelihood of trips between two locations generally decreases due to higher travel costs.

In this research, we propose two region partition strategies: based on election districts and K-means clustering (indicated by *1 in Figure 1). Both strategies can reduce the dimensionality of OD matrices while ensuring that the region partition standards have sound theoretical meanings.

Election districts offer a practical framework for aggregation because they are designed with careful consideration of population distribution (Almeida & Manquinho, 2022). This aligns well with Knowledge 1 that population influences travel demands. Election districts are periodically adjusted to reflect population changes, which makes them a dynamic and relevant basis for aggregation.

Another promising method is using K-means clustering as the aggregation basis, which focuses purely on geographical distance. K-means clustering groups nodes into clusters based on their spatial closeness, which helps to reduce the dimension of OD matrices while preserving the essential spatial relationships among nodes. This method is consistent with Knowledge 1 that distance is a key determinant of travel demands. While travel time plays the most critical role on travel demands, its dynamic nature—affected by traffic conditions and modes of transport—poses significant challenges for clustering methods. Therefore, in this research, we used distance as a stable and reliable measure to aggregate nodes. K-means clustering ensures that nodes within the same cluster are as close to each other as possible, making it easier to analyze travel patterns that are influenced by geographical distance.

4.3 | Multi-graph input GCN module

Knowledge 2: The correlations between each node in a traffic network are dynamic rather than static.

Traffic networks consist of nodes (such as zones, or regions) and edges (roads or pathways). Static graphs assume that the relationships among different nodes remain constant over time. However, this assumption fails to capture the time-dependent nature of OD patterns, leading to less accurate predictions. Therefore, in this research, both static and dynamic graphs were employed as inputs for GCNs (indicated by *2 in Figure 1).

4.3.1 | Static graphs

In this research, two types of static graphs are considered: the adjacency graph and the distance graph; and we used GCNs to capture the static relationships among nodes.

1. Adjacency graph: This graph encodes the connectivity among nodes in the network. The adjacency graph captures whether two nodes are directly connected or not.

$$G_{adj}(i,j) = \begin{cases} 1, & \text{if } i \text{ is adjacent to } j; \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

2. Distance graph: This graph captures the spatial relationships among nodes by applying a thresholded Gaussian kernel weighting function (Shuman et al., 2013; Ye et al., 2023). This method adopts the idea that regions closer to each other tend to have similar traffic patterns, and if the distance between them is greater than a certain threshold, no weight (i.e., 0) is assigned.

$$G_{dis}(i,j) = \begin{cases} \exp\left(-\dfrac{d(i,j)^2}{\sigma^2}\right), & \text{if } \exp\left(-\dfrac{d(i,j)^2}{\sigma^2}\right) \geq \pi; \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Here, $d(i,j)$ refers to the Euclidean distance between nodes i and j. σ^2 denotes the variances of all the distances, and π represents the threshold. In this research, we define the threshold π as the mean value of all the distances.

4.3.2 | Dynamic graphs

In this research, we introduce dynamic graphs to model the time-dependent variations in the network for both origins and destinations and apply layers of BDGCN to simultaneously capture the correlations between them.

The dynamic graphs are constructed based on the relationships between historical OD data. We distinguish the features between origins and destinations and used the Pearson correlation coefficient (PCC) to measure the node correlations for each time step t. The correlation between origin i_1 and i_2 at time step t is defined as

$$G_{dy_t}\left(O_{i_1}, O_{i_2}\right) = PCC\left(\sum_j m_t(i_1,j), \sum_j m_t(i_2,j)\right) \tag{9}$$

Here, $m_t(i,j)$ refers to the flow from origin i to destination j at time step t. For destinations q_1 and q_2, the dynamic correlations are defined similarly:

$$G_{dy_t}\left(D_{q_1}, D_{q_2}\right) = PCC\left(\sum_p m_t(p,q_1), \sum_p m_t(p,q_2)\right) \tag{10}$$

4.4 | Global–local attention

Knowledge 3: Spatial correlations in traffic networks exhibit nonlinear patterns due to the simultaneous influence of both global and local relationships.

Global spatial relationships are the relationships among all the nodes. These correlations consider the overall structure of the network, which is essential for understanding network-wide traffic patterns. Local spatial correlations focus on the interactions between a specific node and its adjacent nodes. This idea is consistent with the vehicle conservation law in traffic flow theory, which states that the number of vehicles exiting a region is influenced by the number of vehicles departing from its surrounding areas. By doing so, we aim to comprehensively capture the spatial correlations of traffic data.

In this research, we propose a novel global–local attention module (indicated by *3 in Figure 1) to model the global–local spatial relationships within the OD data. The concept of the attention mechanism was first proposed by Vaswani et al. (2017) and has since become an important tool in deep learning. It enables the model to filter out and focus on the most relevant information from a large dataset. The larger the attention weight assigned, the more focused it is on its corresponding value. This makes the attention mechanism attractive for people to understand and interpret the model prediction results.

4.4.1 | Global self-attention

Global self-attention is used to capture the spatial correlations among all the nodes, which is crucial for understanding traffic patterns across the entire network. Self-attention measures the relevance between different elements (or tokens) to compute new representations. To achieve this, the OD matrix M is first projected to three matrices query (Q^g), key matrix (K^g), and value (V^g) using Equation (11):

$$\begin{aligned} Q^g &= MW_Q^g \\ K^g &= MW_K^g \\ V^g &= MW_V^g \end{aligned} \tag{11}$$

where W_Q^g, W_K^g, and W_V^g are linear mappings. In practice, the above three matrices are used to calculate attention scores. The dot product of the query (Q^g) and the key (K^g) is first calculated. Then we divide it by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the value (V^g). Here, d_k represents the dimension of query Q^g and key K^g, which serves as a scaling factor (Vaswani et al., 2017). In network-scale traffic analyses, d_k is relatively large, causing the dot products to become excessively large. To mitigate this, $\sqrt{d_k}$ is applied in the denominator. This approach can effectively capture the relationships across the network. The global self-attention can be written as

$$\text{Global Attention } (Q^g, K^g, V^g) = softmax\left(\frac{Q^g {K^g}^T}{\sqrt{d_k}}\right) V^g \tag{12}$$

4.4.2 | VCA

In the modeling and analysis of traffic networks, it is crucial to model both global and local spatial relationships to accurately reflect the nonlinearity of traffic data. Inspired by X. Huang et al. (2023), who developed a local flow attention module based on convolutions and used convolutional kernels to represent neighborhood ranges, we have developed a novel local attention module, called VCA, which is more suitable for graph-based data.

Compared to existing local attention modules, which use a single OD matrix (Feng & Tassiulas, 2022), our VCA computes attention scores using both inflow and outflow matrices. The vehicle conversation law in traffic flow theory states that the number of vehicles exiting a region is influenced by the number of vehicles entering from surrounding areas. This principle forms the foundation of the VCA, allowing it to determine how each node receives its incoming traffic based on the availability and conditions of its neighbors. This approach ensures that traffic flow dynamics are captured more realistically. Specifically, for a target node j, we first compute both the inflow matrix of it and the outflow matrix of its neighborhood nodes $N(j)$:

$$\begin{aligned} M_{in}(j) &= \sum_i m(i,j) \\ M_{out}(N_{(j)}) &= \sum_{k \in N(j)} \sum_q \mathrm{A}_{adj}(k,q) \cdot m(k,q) \end{aligned} \tag{13}$$

Here, $m(i,j)$ represents the flow from origin i to destination j, which is the element in the OD matrix. $M_{in}(j)$ aggregates all the inflow features for node j. $M_{out}(N_{(j)})$ denotes the outflows from neighboring nodes. $A_{adj}(k,i)$ represents the connectivity between node k and q, which is the element in the adjacency matrix. Then, the matrices query Q^l, key K^l, and value V^l for node j is obtained using Equation (14):

$$\begin{aligned} Q^l(j) &= (M_{in}(j))W_Q^l \\ K^l(N_{(j)}) &= (M_{out}(N_{(j)}))W_K^l \\ V^l(N_{(j)}) &= (M_{out}(N_{(j)}))W_V^l \end{aligned} \tag{14}$$

where W_Q^l, W_K^l, and W_V^l are linear mappings. The query Q^l is naturally linked to the inflow, as it serves as the

starting point for tracing back its origins. The key K^l and value V^l are linked to the outflow from neighboring nodes, representing the "supply' of traffic available to meet the inflow 'demand." Then, the VCA weights were calculated by Equation (15):

$$VCA(Q^l, K^l, V^l) = softmax(w_D(i,j) \odot \left(\frac{(Q^l(j)K^l(N_{(j)})^T)}{\sqrt{d_k}} \right) V^l(N_{(j)}) \quad (15)$$

Here, $\odot$ denotes the element-wise multiplication. The calculation of VCA is similar as in the global self-attention but integrates a distance weight $w_D(i,j)$. $w_D(i,j)$ aims to emphasize the importance of spatial proximity between nodes, and it takes the same value as it is in the distance matrix $G_{dis}(i,j)$.

$$w_D\ (i,j) = G_{dis}\ (i,j) = \begin{cases} \exp\left(-\frac{d(i,j)^2}{\sigma^2}\right), & \text{if } \exp\left(-\frac{d(i,j)^2}{\sigma^2}\right) \geq \pi; \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

The attention score is generated by the interaction between the query Q^l and key K^l incorporating the distance weight $w_D(i,j)$. It quantifies the relevance of each neighbor's outflow to the current node's inflow while ensuring that the VCA reflects both the flow dynamics and the physical relationships between nodes. This approach aligns with the core principles of attention mechanisms. Moreover, the VCA mechanism enables the model to effectively capture complex spatial node correlations and align with real-world traffic flow dynamics, thereby enhancing its interpretability.

4.5 | MLR for including external factors

External factors have significant impacts on traffic demand, thus it is essential to consider those factors in the ODP model.

In this research, we propose using MLR to model the relationship between OD data and external factors (indicated by *4 in Figure 1). A commonly used MLR formula in vector form is given as follows:

$$Y = AX + BZ + \epsilon \quad (17)$$

where Y is the dependent variable, X and Z are independent variables. A, B are the parameters to be estimated,

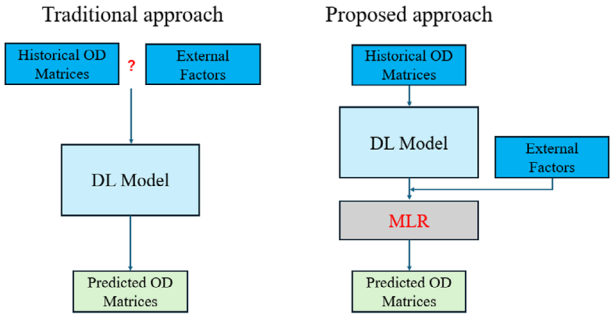


FIGURE 2 Comparison of the traditional approach and proposed approach for integrating external factor data. DL, deep learning; MLR, multi-linear regression; OD, origin–destination.

and ϵ is the error term. We modify this MLR expression by allowing the first term to have nonlinear relations to y:

$$Y = f(X) + BZ + \epsilon \quad (18)$$

Here, $f(\cdot)$ denotes complex functions such as neural networks. Therefore, in the ODP problem, replacing Y as the predicted OD matrix $\hat{M}$, X as the historical OD matrices M, and Z as the external factor matrix E, the following Equation (19) can be obtained:

$$\hat{M} = f(M) + BE + \epsilon \quad (19)$$

This external factor inclusion approach enables us to understand the relationship between OD matrices and external factors. A comparison between the traditional and the proposed approach is given in Figure 2. Moreover, the effects of individual external factor E^i ($E^i \in E$) on traffic data can be quantified through calculating a ratio between the output of neural network $f(M)$ and individual external factor B^iE^i.

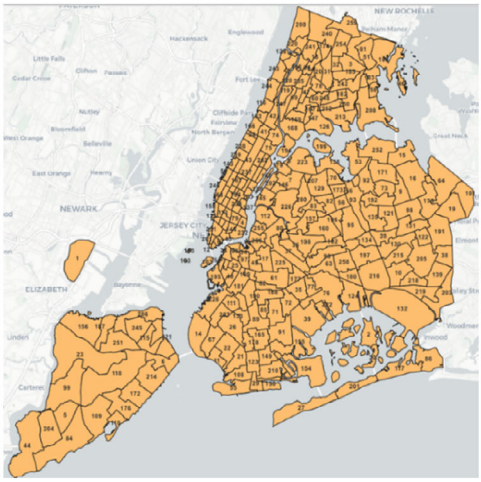
5 | EXPERIMENTS

In this section, we describe the settings of the experiments conducted on real-world datasets to validate the efficiency of our proposed method. Thirteen baselines and four evaluation metrics were adopted to assess the prediction performance.

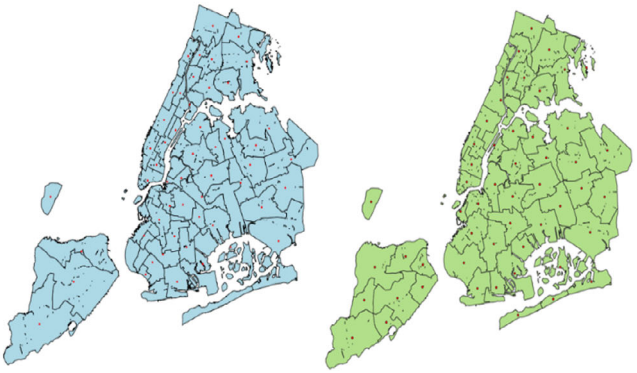
5.1 | Data description

5.1.1 | OD data

To assess the performance of our proposed ODP framework, extensive experiments were conducted by applying two real-world datasets:

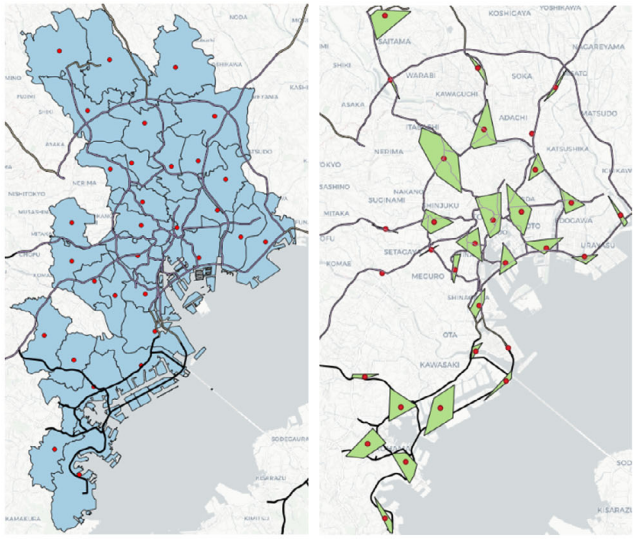


(a) NYC yellow taxi zones.



(b) NYC taxi zone partition by election districts (left) and K-means clustering (right).

(c) Tokyo MEX ETC toll gates.



(d) Tokyo MEX toll gate partition by election districts (left) and K-means clustering (right).

FIGURE 3 Visualizations of the two datasets and the two region partitioning methods. ETC, electronic toll collection; NYC, New York City; Tokyo MEX, Tokyo Metropolitan Expressway.

FIGURE 3 Continued

1. New York yellow taxi data (NYC Taxi and Limousine Commission (TLC) trip record data. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page) contains trip records from 263 taxi service zones across New York City (NYC). These zones were defined by the TLC based on taxi activity, with each zone assigned a unique number. Figure 3a shows the division of these taxi zones. Each taxi trip record contains key information such as pick-up and drop-off taxi zone IDs, time, and so forth. For this study, we selected data spanning from January to June 2023.
2. Tokyo Metropolitan Expressway (MEX) ETC data include trip records for 495 toll gates on the expressway network. Figure 3c illustrates the locations of these toll gates. The data, collected via the MEX Electronic Toll Collection (ETC) system, provides information on individual trips, such as time-stamped records of vehicle entry and exit toll gates. We used data collected from April to December 2019.

Both datasets share common characteristics: They feature specific high-demand traffic regions, such as Manhattan in NYC and central Tokyo, and exhibit distinct ST commuting behavior, making them particularly well-suited for evaluating ST deep learning models. On the other hand, the key differences between the datasets lie in the transportation modes they represent and their structure. The NYC yellow taxi dataset reflects taxi usage and is zone-based, aggregated within predefined taxi zones. Tokyo MEX ETC dataset captures highway vehicle trips and is point-based, centered on isolated ETC toll gate locations.

To prepare the datasets for OD analysis, we partitioned both the NYC and Tokyo MEX networks based on election

districts and K-means clustering. The resulting partitions are illustrated in Figure 3b,d. The OD data were aggregated for each defined district or cluster, with an hourly temporal resolution. It is important to note that the input data used in this study represent only a subset of the total OD in each region. For NYC, the analysis is based on taxi data, excluding other modes of transportation. For Tokyo, the data are limited to traffic on the expressways, which do not capture OD flows on local roads. These limitations should be considered when interpreting the results.

In NYC, we used 51 election districts as defined by the NYC Districting Commission (2022). Additionally, the dataset includes taxi data for Newark Liberty International Airport, which lies outside NYC, resulting in an OD matrix dimension of 52 × 52 for each hour. For the Tokyo metropolitan area, we considered the election districts for Tokyo, Saitama, Chiba, and Kanagawa Prefectures, totaling 80 districts (Nishikawa, 2022). However, not all districts are located along the MEX. Therefore, we extracted only those districts containing at least one ETC toll gate, resulting in 31 districts. This led to an OD matrix dimension of 31× 31 per hour.

To enable a fair comparison between the two regional partitioning methods, we chose 52 and 31 clusters for K-means clustering for the NYC and Tokyo datasets, respectively. It is worth noting that in the Tokyo MEX dataset, the K-means clustering method aggregated isolated points (toll gates) rather than contiguous zones. To address this, cluster boundaries were defined using convex hulls to encompass each set of toll gates.

5.1.2 | External factor data

In this research, three types of external factors were considered: weekends, holidays, and weather. Since the traffic patterns are generally different during weekdays, weekends, and holidays, we use one-hot encoding to distinguish the day types:

$$E_T^W = \begin{cases} 1, & \text{if } T \text{ is weekend;} \\ 0, & \text{if } T \text{ is weekday} \end{cases} \tag{20}$$

$$E_T^H = \begin{cases} 1, & \text{if } T \text{ is holiday;} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

Furthermore, many research works have proven that precipitation has a significant impact on traffic demands (Chung et al., 2005; Romanowska & Budzyński, 2022). Therefore, we selected precipitation as a representation of weather external factors. The precipitation data for NYC was obtained from visual crossing (historical weather data for NYC. https://www.visualcrossing.com/weather-history/New+York+City%2CUSA), and the precipitation data for Tokyo was provided by the Japan Weather Association. Regarding the processing of precipitation data, we follow the processing method in Qi et al. (2024), which transformed continuous precipitation values into six discrete categorical levels according to the rainfall intensities. The details are shown in Equation (22), and the unit of precipitation is [inches/h]. The use of continuous values may result in the issue that small variations in precipitation have little impact on traffic data. By introducing discrete categories, changes in precipitation are amplified, which can accelerate the model training process (Qi et al., 2024).

$$E_T^P = \begin{cases} 0, & precipitation = 0; \\ 1, & 0 < precipitation < 0.01; \\ 2, & 0.01 < precipitation < 0.02; \\ 3, & 0.02 < precipitation < 0.1; \\ 4, & 0.1 < precipitation < 0.3; \\ 5, & precipitation > 0.3 \end{cases} \tag{22}$$

5.2 | Evaluation metrics

In this research, four indicators were applied to evaluate the performance of the models: root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and mean structural similarity index (MSSIM). The formulations for the first three metrics are given in Equations (23) to (25):

$$\text{RMSE}\left(M, \hat{M}\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(m(i,j) - \hat{m}(i,j)\right)^2} \tag{23}$$

$$\text{MAE}\left(M, \hat{M}\right) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|m(i,j) - \hat{m}(i,j)\right| \tag{24}$$

$$\text{MAPE}\left(M, \hat{M}\right) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|\frac{m(i,j) - \hat{m}(i,j)}{m(i,j)}\right| \tag{25}$$

Behara et al. (2022) highlighted the importance of including both the value differences and the structural similarities between the OD matrices when comparing different OD pairs. Traditional measures that compare OD matrices purely based on deviations of individual OD flows may yield less meaningful prediction results when they fail to consider structural differences in the OD matrices. The structural similarity index (SSIM) computes statistics within a local window that slides across the entire OD matrix. The SSIM evaluates the structural similarity between those local windows from both OD matrices. The MSSIM is computed by averaging the local SSIM values for

all sliding windows. The MSSIM between the OD matrices M and $\hat{M}$ is calculated by Equation (26). For more details, we refer to Behara et al. (2022).

$$\text{MSSIM}\left(M,\hat{M}\right) = \frac{1}{S}\sum_{i=1}^{s}\frac{\left(2\mu_{M_s}\mu_{\widehat{M_s}}+C_1\right)\left(2\sigma_{M_s\widehat{M_s}}+C_2\right)}{\left(\mu_{M_s}{}^2+\mu_{\widehat{M_s}}{}^2+C_1\right)\left(\sigma_{M_s}{}^2+\sigma_{\widehat{M_s}}{}^2+C_2\right)} \quad (26)$$

Here, M_s and $\widehat{M_s}$ represent the sub-matrices within local windows in both M and $\hat{M}$. μ_{M_s} and $\mu_{\widehat{M_s}}$ denotes the means, $\sigma_{M_s}{}^2$ and $\sigma_{\widehat{M_s}}{}^2$ are the local variances, $\sigma_{M_s\widehat{M_s}}$ means the local covariance of M_s and $\widehat{M_s}$. C_1 and C_2 are constants to stabilize the division when denominators are close to zero. Unlike other matrices, a larger value of MSSIM indicates the predicted OD matrices are closer to the real OD matrices.

5.3 | Baseline models

The performance of our proposed KE-H-GNN was compared with the following baseline models:

1. HA-1: HA-1 performs prediction by averaging the OD matrices of the previous time steps.
2. HA-2: HA-2 performs prediction by averaging the OD matrices from: the same time on previous occurrences of the same day of the week and the previous time step.
3. Gated recurrent unit network (GRU): A special variant of RNN with a gating mechanism, which has been widely applied in time-series prediction.
4. LSTM: Another special variant of RNN, which also has been widely used in time-series prediction.
5. Attention-based LSTM (At-LSTM); X. Zhang et al., 2019): Combined LSTM network with an attention mechanism.
6. STGCN (Yu et al., 2018): A model based on gated convolution and graph convolution.
7. Diffusion convolutional RNN (DCRNN; Yaguang Li et al., 2018): Combined diffusion convolution and sequence modeling using GRUs. It is a benchmark model in ST traffic prediction.
8. Attention-based STGCN (ASTGCN; Guo et al., 2019): Integrated ST graph convolutions with attention mechanisms, which is another benchmark model in ST traffic prediction.
9. ST transformer network (STTN; Xu et al., 2020): A transformer-based model. STTN consists of a series of ST blocks and one prediction layer. Each ST block contains one spatial transformer and one temporal transformer to capture the ST relationship simultaneously.
10. MPGCN (Shi et al., 2020): MPGCN first applies LSTM to extract temporal features. Then it captures the spatial correlations of nodes by three branches of BDGCN, with adjacency graph, PoI similarity graph, and dynamic cosine similarity graph.
11. OD convolutional recurrent network (ODCRN; Jiang et al., 2021): A model that incorporates an encoder–decoder structure with two types of graph convolution units. The first utilizes static graph as auxiliary input, while the second employs a dynamic graph constructor to generate dynamic graphs based on current observations.
12. Attention-enhanced graph convolutional LSTM network (AGC-LSTM; Y. Zhang et al., 2024): The model integrates multi-head attention mechanism with GCN and LSTM.
13. GraphResLSTM (J. Zhang & Hu, 2024): A hybrid model based on the architecture of residual neural network. It employed GCN and LSTM to model the ST dependencies.

5.4 | Experiment settings

In this research, both datasets were separated into three parts: the first 60% for training, the following 20% for validation, and the last 20% for testing.

For all the baseline models, the hyperparameters were chosen through trial and error. For a fair comparison, we fixed the hidden dimensions to 64 for our KE-H-GNN (including GCN, BDGCN, and LSTM) as well as for other baseline models that require hidden dimensions. The number of layers for LSTM was set to 1, and the number of layers for GCN and BDGCN was set to 2. The Chebyshev order of the GCN and BDGCN (and the hop size for BDGCN) was set to 2. Adam optimizer was used to train the KE-H-GNN model. Standard normalization is applied to scale the historical OD values within [−1,1] to increase the convergence speed during the training process.

Moreover, the batch size was set to 32. To prevent overfitting, we experimented with dropout rates within {0.1, 0.2, 0.3, 0.4, 0.5}. Through these experiments, we found that the best value was 0.3. Additionally, the learning rates were tested within (0.1, 0.01, 0.001, 0.0001). Mean square error was applied as the loss function. Moreover, as mentioned previously, small values greatly influence the prediction results, and those small values have little practical meaning. Therefore, for NY taxi data, only the values equal or greater than 1 were included for evaluation; and for MEX ETC data, values equal or greater than 10 were included for evaluation since the traffic demands are rather large in

this case. The training process is early stopped within 10 epochs.

All experiments were implemented in PyTorch and repeated five times on a single NVIDIA GeForce RTX 4090 GPU and Intel(R) Core(TM) i9-14900HX CPU. We report the best results for each baseline model.

6 | RESULTS AND DISCUSSIONS

In this section, we present and analyze the experimental results. For all Sections from 6.1 to 6.3, OD data were aggregated using the K-means clustering approach. On the other hand, in Section 6.4, we compare the effectiveness of the proposed two meaningful region partitioning strategies.

6.1 | Overall performance

In this subsection, we report the performance of all methods. Table 1 compares the overall performance between the baselines and the proposed KE-H-GNN model for single-step ODP tasks for both datasets. The best values were highlighted in bold. Here, we also show a variant of KE-H-GNN without external factors, namely, knowledge-enhanced ST graph neural networks (KE-H-GNN). Figures 4 and 5 show the results of stepwise predictions for the NYC yellow taxi and Tokyo MEX ETC datasets, respectively. The time units for the prediction steps in Figures 4 and 5 are hourly. We note that the MAPE of HA-1 is significantly larger than the other baseline models for both datasets. Therefore, we took smaller ranges for the y-axis to better visualize the results for other models, which resulted in it falling outside the figures.

Regarding the next-step short-term (1 h) prediction, the proposed KE-H-GNN outperforms all the baseline models compared. Moreover, KE-GNN achieves the second-best performance. Compared to the reference model ODCRN, which performed the best among all the existing models, for the NYC yellow taxi dataset, our proposed KE-H-GNN achieved the following improvements: 6.90% in average RMSE, 1.83% in average MAE, 16.79% in average MAPE, and 0.12% in average MSSIM. Regarding the Tokyo MEX ETC dataset, compared to the reference model ODCRN, we also improved the average RMSE by 1.89%, average MAE by 2.70%, average MAPE by 2.65%, and average MSSIM by 0.72%. Moreover, we found that HA-1 has the lowest prediction accuracy, which is not surprising since it cannot capture the ST dependencies within the OD data, compared to other deep learning-based models. However, compared to HA-1, HA-2 achieved relatively higher performance as it considers average OD patterns from the same time on the same day of the previous weeks, which represents a form of temporal dependency modeling. Among the deep learning-based models, the ST convolution-based models, including STGCN and ASTGCN outperform RNN-based models including GRU, LSTM, and At-LSTM because RNN-based models lack the ability to retain spatial relationships. Furthermore, the performance of STGCN and DCRNN are competitive in terms of value-based deviation metrics (RMSE, MAE, MAPE). However, in terms of MSSIM, both models do not perform as well. This again highlights the importance of considering both value deviations and structural similarities between the predicted and true OD matrices when evaluating the performance of ODP models. More recent models, including graph convolution-based model (MPGCN, ODCRN, AGC-LSTM, GraphResLSTM) and transformer-based model (STTN) perform better than convolution-based models.

Regarding mid- to long-term predictions, all the models show a decreasing tendency in performance as the prediction step increases. Because as predictions continue, the predicted OD from the previous step is used for the following steps, causing errors to accumulate over time. Under this situation, the proposed KE-H-GNN model consistently outperforms the baseline models across all the time steps. This is evident in both value-based metrics and structural metrics. As shown in both figures, although the performance differences between KE-H-GNN and some models, such as DCRNN and ODCRN, are relatively small in the initial steps, these differences become more pronounced as the prediction step increases. This again highlights the effectiveness of the proposed model for mitigating error accumulation in long-term ODP. Interestingly, HA-2 exhibits a sharp increase in MAPE and MSSIM starting from prediction Step 3. However, its RMSE and MAE increase rather slowly, compared to the other two metrics, even outperforming some deep learning models. Also, compared to RNN-only models (GRU, LSTM), the introduction of attention mechanism (At-LSTM) and transformer-based architecture (STTN) enables the models to mitigate the issue of long-term error accumulation. Moreover, we can observe that the MAPE of STGCN and ASTGCN increase sharply as the prediction steps progress. This may be because STGCN and ASTGCN were designed based on both gated convolutions and graph convolutions; however, gated convolutions are unsuitable for data aggregated on irregular shape zones (e.g., clusters). On the other hand, the error accumulation rate is also high for ODCRN and DCRNN, despite their strong short-term performance. In contrast, GCN and LSTM-based models, including MPGCN, AGC-LSTM, and GraphResLSTM, demonstrate more stable performance in mid- to long-term predictions. However, they still underperform, compared to the proposed KE-H-GNN, since KE-H-GNN integrates knowledge from traffic engineering. KE-H-GNN incorporates not only global self-attention but also localized VCA. VCA was designed based on traffic flow theory, which is

TABLE 1 Results for single-step origin–destination prediction.

	New York City (NYC) yellow taxi					Tokyo Metropolitan Expressway (MEX) electronic toll collection (ETC)				
Baselines	**Root mean square error (RMSE; veh/h)**	**Mean absolute error (MAE; veh/h)**	**Mean absolute percentage error (MAPE; %)**	**Mean structural similarity index (MSSIM)**	**Run time (s)**	**RMSE (veh/h)**	**MAE (veh/h)**	**MAPE (%)**	**MSSIM**	**Run time (s)**
HA-1	11.29–	0.909–	8.39–	0.9800–	–	75.10–	15.33–	68.13–	0.9074–	–
HA-2	4.18–	0.340-	4.14–	0.9888-	–	38.96–	8.56–	21.29–	0.9237–	–
GRU	4.32 ± 0.03	0.376 ± 0.02	3.09 ± 0.05	0.9896 ± 0.0001	1668 ± 147	41.20 ± 1.21	9.56 ± 0.09	14.84 ± 0.08	0.9161 ± 0.0010	651 ± 150
LSTM	4.23 ± 0.09	0.339 ± 0.01	3.42 ± 0.14	0.9877 ± 0.0011	1853 ± 251	40.71 ± 1.45	9.52 ± 0.03	15.02 ± 0.04	0.9167 ± 0.0011	981 ± 185
At-LSTM	3.70 ± 0.02	0.338 ± 0.01	2.69 ± 0.01	0.9916 ± 0.0002	1441 ± 226	38.52 ± 0.17	9.41 ± 0.09	15.53 ± 0.05	0.9283 ± 0.0006	461 ± 67
STGCN	3.64 ± 0.14	0.346 ± 0.03	2.92 ± 0.36	0.9910 ± 0.0022	1356 ± 241	34.87 ± 2.19	9.02 ± 0.73	14.97 ± 0.44	0.9235 ± 0.0007	645 ± 64
ASTGCN	3.54 ± 0.13	0.369 ± 0.01	2.81 ± 0.10	0.9933 ± 0.0020	1497 ± 247	31.46 ± 1.17	8.33 ± 0.37	14.59 ± 0.94	0.9316 ± 0.0068	698 ± 158
STTN	3.12 ± 0.20	0.271 ± 0.03	2.66 ± 0.30	0.9961 ± 0.0008	2519 ± 344	28.52 ± 0.43	7.64 ± 0.22	13.55 ± 1.31	0.9252 ± 0.0018	2264 ± 498
DCRNN	2.97 ± 0.19	0.273 ± 0.01	2.52 ± 0.04	0.9942 ± 0.0011	1492 ± 134	27.50 ± 1.15	7.35 ± 0.01	13.82 ± 0.06	0.9321 ± 0.0006	914 ± 53
MPGCN	3.13 ± 0.18	0.275 ± 0.01	2.56 ± 0.05	0.9980 ± 0.0001	1821 ± 207	27.03 ± 0.20	7.50 ± 0.03	13.72 ± 0.06	0.9326 ± 0.0012	2875 ± 596
ODCRN (reference model)	2.90 ± 0.12	0.273 ± 0.02	2.68 ± 0.27	0.9972 ± 0.0003	7011 ± 395	26.42 ± 0.24	7.40 ± 0.10	13.60 ± 0.05	0.9294 ± 0.0026	3474 ± 264
AGC-LSTM	3.26 ± 0.13	0.291 ± 0.01	2.57 ± 0.03	0.9976 ± 0.0002	1324 ± 243	33.78 ± 0.80	8.61 ± 0.07	14.00 ± 0.19	0.9240 ± 0.0044	746 ± 159
GraphResLSTM	3.25 ± 0.05	0.310 ± 0.01	2.49 ± 0.05	0.9979 ± 0.0001	2347 ± 362	32.12 ± 0.04	8.41 ± 0.02	14.17 ± 0.05	0.9235 ± 0.0010	2533 ± 109
KE-GNN (ours)	2.88 ± 0.06	0.284 ± 0.01	2.37 ± 0.06	0.9981 ± 0.0002	1653 ± 286	26.50 ± 0.09	7.32 ± 0.10	13.50 ± 0.13	0.9334 ± 0.0023	1870 ± 150
KE-H-GNN (ours)	**2.70 ± 0.04** (+6.90%)	**0.268 ± 0.02** (+1.83%)	**2.23 ± 0.11** (+16.79%)	**0.9984 ± 0.0003** (+0.12%)	1880 ± 312 (+73.19%)	**25.92 ± 0.10** (+1.89%)	**7.20 ± 0.03** (+2.70%)	**13.24 ± 0.24** (+2.65%)	**0.9361 ± 0.0004** (+0.72%)	2003 ± 171 (+42.34%)

Abbreviations: AGC-LSTM, attention-enhanced graph convolutional long short-term memory network; At-LSTM, attention-based Long short-term memory; ASTGCN, attention-based spatial–temporal graph convolutional network; DCRNN, diffusion convolutional recurrent neural network; GRU, gated recurrent unit network; ETC, electronic toll collection; HA-1, historical average 1; HA-2, historical average 2; KE-H-GNN, knowledge-enhanced hybrid spatial–temporal graph neural network; LSTM, long short-term memory; MPGCN, multi-perspective graph convolutional network; ODCRN, origin–destination convolutional recurrent network; STGCN, spatial–temporal graph convolutional network; STTN, spatial–temporal transformer network.

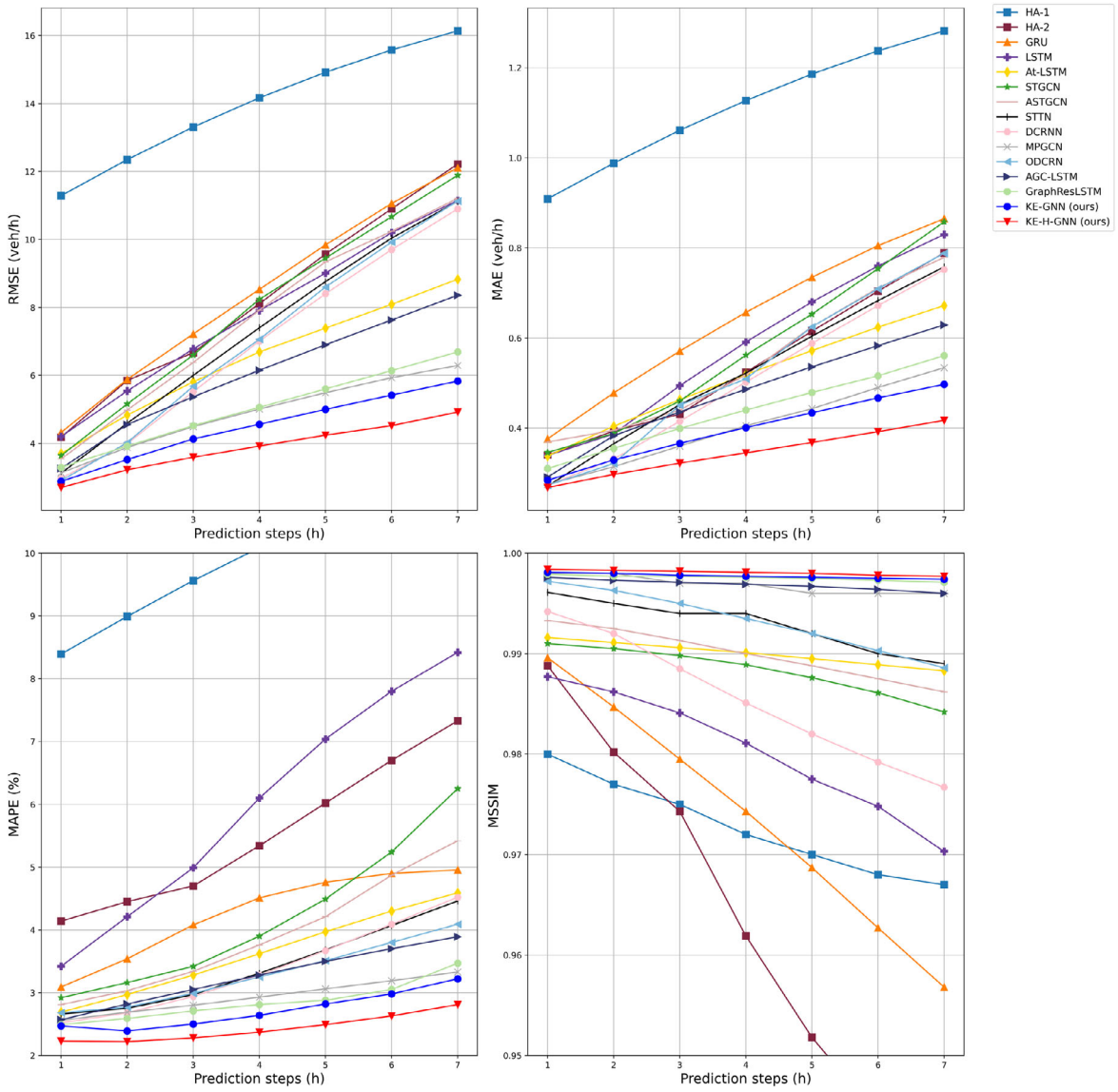


FIGURE 4 Results for stepwise origin–destination (OD) prediction (ODP; NYC yellow taxi). AGC-LSTM, attention-enhanced graph convolutional long short-term memory network; At-LSTM, attention-based Long short-term memory; ASTGCN, attention-based spatial–temporal graph convolutional network; DCRNN, diffusion convolutional recurrent neural network; GRU, gated recurrent unit network; ETC, electronic toll collection; HA-1, historical average 1; HA-2, historical average 2; KE-H-GNN, knowledge-enhanced hybrid spatial–temporal graph neural network; LSTM, long short-term memory; MAE, mean absolute error; MAPE, mean absolute percentage error; MPGCN, multi-perspective graph convolutional network; MSSIM, mean structural similarity index; ODCRN, origin–destination convolutional recurrent network; RMSE; STGCN, spatial–temporal graph convolutional network; STTN, spatial–temporal transformer network; Tokyo MEX, Tokyo Metropolitan Expressway.

closely related to real-world traffic phenomena. By introducing VCA, the model's focus can be narrowed to the neighboring nodes of specific nodes, helping it to more efficiently capture the node correlations.

Finally, from the runtime results, we observe that the runtime of the proposed KE-H-GNN is highly advantageous, compared to more advanced models such as STTN, MPGCN, ODCRN, and GraphResLSTM. Particularly, while ODCRN achieves competitive performance in the single-step ODP task, it requires extensive training time. Additionally, although GraphResLSTM incorporates GCN and LSTM in its architecture to model ST dependencies, the absence of traffic engineering domain knowledge leads to slower convergence during training. In contrast, our proposed KE-H-GNN requires significantly less training time, achieving a 73.19% reduction for the NYC yellow taxi dataset and 42.34% reduction for Tokyo MEX ETC dataset, compared to the reference model. This highlights its potential for practical adoption in real-world situations.

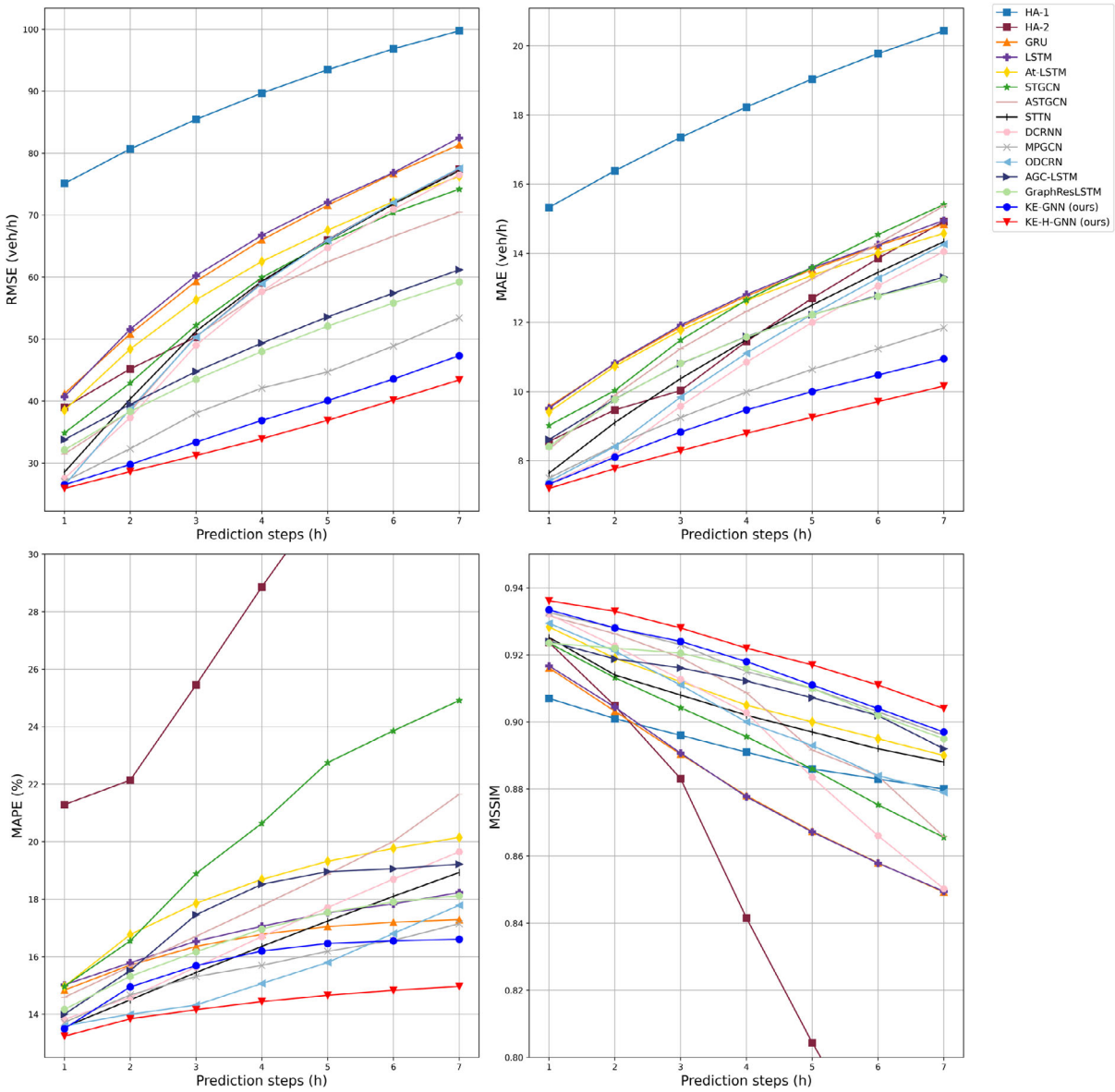


FIGURE 5 Results for stepwise ODP (Tokyo Metropolitan Expressway [MEX] electronic toll collection [ETC]). AGC-LSTM, attention-enhanced graph convolutional long short-term memory network; At-LSTM, attention-based Long short-term memory; ASTGCN, attention-based spatial–temporal graph convolutional network; DCRNN, diffusion convolutional recurrent neural network; GRU, gated recurrent unit network; HA-1, historical average 1; HA-2, historical average 2; KE-GNN; KE-H-GNN, knowledge-enhanced hybrid spatial–temporal graph neural network; LSTM, long short-term memory; MAE, mean absolute error; MAPE, mean absolute percentage error; MPGCN, multi-perspective graph convolutional network; MSSIM, mean structural similarity index; ODCRN, origin–destination convolutional recurrent network; RMSE; STGCN, spatial–temporal graph convolutional network; STTN, spatial–temporal transformer network.

6.2 | Ablation study

6.2.1 | Ablation study on different model components

In this subsection, we analyze the impact of different model components by examining their contributions to overall performance. We evaluate the effects of removing key components, such as the dynamic graph, global–local attention, and external factors. These variants are represented as "w/o dynamic graph, global–local attention, external factors," "w/o global–local attention," "w/o dynamic graph," and "w/o external factors," alongside the proposed KE-H-GNN. Table 2 presents the results of these variants. The best values were highlighted in bold.

From Table 2, we confirm that the proposed KE-H-GNN model achieves the best performance across both datasets, indicating that the removal of any module results

TABLE 2 Results of the ablation study.

Baselines	NYC yellow taxi				Tokyo MEX ETC			
	RMSE (veh/h)	MAE (veh/h)	MAPE (%)	MSSIM	RMSE (veh/h)	MAE (veh/h)	MAPE (%)	MSSIM
W/o dynamic graph, global–local attention, external factors	3.51 ± 0.05	0.314 ± 0.01	2.67 ± 0.08	0.9881 ± 0.0003	30.05 ± 1.25	8.26 ± 0.22	14.57 ± 0.33	0.9276 ± 0.0028
W/o global–local attention	3.18 ± 0.04	0.297 ± 0.01	2.49 ± 0.02	0.9977 ± 0.0004	27.16 ± 0.36	7.83 ± 0.02	14.44 ± 0.48	0.9335 ± 0.0004
W/o dynamic graph	3.12 ± 0.05	0.297 ± 0.01	2.40 ± 0.06	0.9975 ± 0.0005	26.37 ± 0.17	7.44 ± 0.04	14.15 ± 0.10	0.9341 ± 0.0001
W/o external factors (Reference Model)	2.88 ± 0.06	0.284 ± 0.01	2.37 ± 0.06	0.9981 ± 0.0002	26.50 ± 0.09	7.32 ± 0.10	13.50 ± 0.13	0.9334 ± 0.0023
KE-H-GNN (ours)	**2.70 ± 0.04** (+6.25%)	**0.268 ± 0.02** (+5.63%)	**2.23 ± 0.11** (+5.91%)	**0.9984 ± 0.0003** (+0.03%)	**25.92 ± 0.10** (+2.19%)	**7.20 ± 0.03** (+3.23%)	**13.24 ± 0.24** (+1.93%)	**0.9361 ± 0.0004** (+0.29%)

Abbreviation: KE-H-GNN, knowledge-enhanced hybrid spatial–temporal graph neural networks.

in a degradation of the prediction performance. The model without all the components ("w/o dynamic graph, global–local attention, external factors'") performs the worst, especially in terms of MSSIM, as it relies solely on a simple LSTM-GCN with static graphs. This simple structure may limit its ability to capture important spatial information among nodes.

Furthermore, it can be observed that, among the three modules, global–local attention contributes the most to model performance, as the removal of it leads to the most significant decrease in prediction accuracy. This again highlights that the proposed global–local attention module is particularly effective in capturing complex spatial dependencies. Additionally, compared to "w/o global–local attention," the performance of "w/o dynamic graph" shows a slight improvement, suggesting that while the dynamic graph is important for modeling traffic dynamics, it is somewhat less critical than global–local attention. The reference model "w/o external factors" performs the closest to the full model, particularly on the NYC yellow taxi dataset. However, its performance slightly decreases on the Tokyo MEX ETC dataset, indicating that external factors have a larger impact on this dataset.

6.2.2 | Contributions of Individual graphs

In this subsection, we compare the contributions of individual graphs by removing each of them. These variants are denoted as 'w/o adjacent graph', 'w/o distance graph', 'w/o dynamic graph', along with the proposed KE-H-GNN. Table 3 presents the results of these variants. The best values were highlighted in bold.

From Table 3, we confirmed that the proposed KE-H-GNN model performs the best across both datasets, indicating that the removal of each graph reduces the prediction accuracy. For the NYC yellow taxi dataset, we observed that the "w/o dynamic graph" performs the worst, indicating that dynamic graph performs the most important role in this dataset. The performance of "w/o adjacency graph" variant is slightly better than that of the "w/o dynamic graph" but worse than "w/o distance graph." This indicates that the adjacency graph contributes more to the model performance, compared to the distance graph. These findings suggest that temporal traffic demand patterns in the NYC yellow taxi dataset are highly dynamic and are effectively captured by the dynamic graph.

Interestingly, for the Tokyo MEX ETC dataset, the "w/o adjacency graph" performs the worst, highlighting the importance of the adjacency graph in this dataset. Meanwhile, the performance of the "w/o dynamic graph" variant is slightly better than that of the "w/o adjacency graph" but worse than "w/o distance graph," suggesting

TABLE 3 Effects of individual graphs.

Baselines	NYC yellow taxi				Tokyo MEX ETC			
	RMSE (veh/h)	MAE (veh/h)	MAPE (%)	MSSIM	RMSE (veh/h)	MAE (veh/h)	MAPE (%)	MSSIM
W/o dynamic graph	3.12 ± 0.05	0.297 ± 0.01	2.40 ± 0.06	0.9975 ± 0.0005	26.37 ± 0.17	7.44 ± 0.04	14.15 ± 0.10	0.9341 ± 0.0001
W/o adjacent graph	3.08 ± 0.06	0.300 ± 0.01	2.53 ± 0.04	0.9980 ± 0.0001	27.49 ± 0.44	7.65 ± 0.08	13.76 ± 0.15	0.9298 ± 0.0007
W/o distance graph (reference model)	2.81 ± 0.04	0.278 ± 0.01	2.38 ± 0.02	0.9981 ± 0.0001	26.13 ± 0.06	7.34 ± 0.05	13.51 ± 0.16	0.9281 ± 0.0003
KE-H-GNN (ours)	**2.70 ± 0.04** (+3.91%)	**0.268 ± 0.02** (+3.60%)	**2.23 ± 0.11** (+6.30%)	**0.9984 ± 0.0003** (+0.03%)	**25.92 ± 0.10** (+0.80%)	**7.20 ± 0.03** (+1.91%)	**13.24 ± 0.24** (+2.00%)	**0.9361 ± 0.0004** (+0.86%)

Abbreviation: KE-H-GNN, knowledge-enhanced hybrid spatial–temporal graph neural networks.

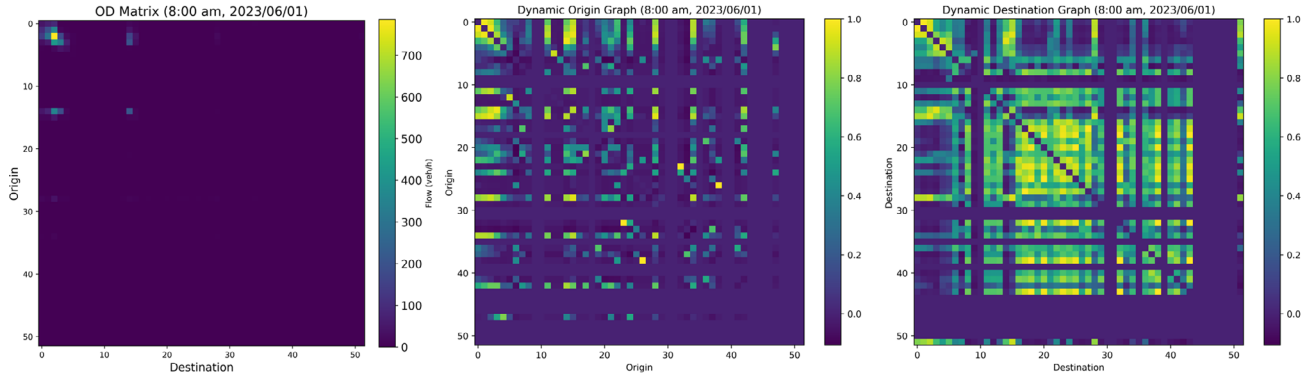
that spatial connectivity plays a dominant role in the Tokyo MEX ETC dataset.

To further demonstrate the temporal dynamics captured by the proposed KE-H-GNN, we visualized the OD matrix and dynamic graphs for both datasets in Figure 6. For the NYC yellow taxi dataset, we selected two distinct scenarios: a weekday morning peak (8:00 a.m., June 1, 2023, and a Sunday early morning (5:00 a.m., June 26, 2023). Similarly, for the Tokyo MEX ETC dataset, we selected two scenarios: a weekday morning peak (8:00 a.m., October 15, 2019) and a holiday midnight (1:00 a.m., December 28, 2019). These visualizations reveal notable differences in OD patterns and node correlations over different periods. In the NYC yellow taxi example, the early morning (low demand) scenario exhibits sparser node correlations, corresponding to reduced travel activity during this time. Meanwhile, in the Tokyo MEX ETC dataset, strong correlations are predominantly observed around high-demand nodes, suggesting the effectiveness of dynamic graphs in capturing temporal variations in node correlations.

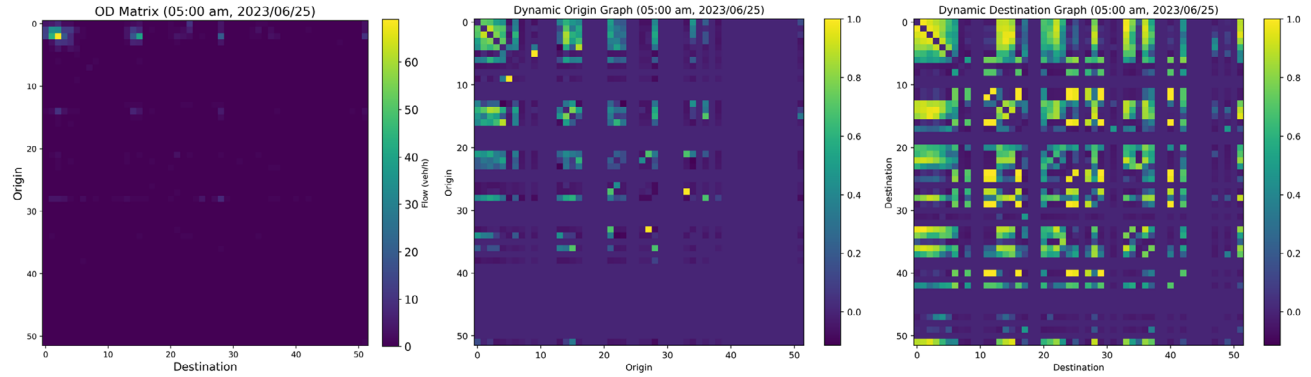
6.3 | Efficiency analyses of MLR

In this subsection, we demonstrate the efficiency of the proposed MLR module by comparing it with the existing concatenation-based method for including external factors, such as Zhang et al. (2021) and Qi et al. (2024). As shown previously in Figure 2, the comparative model is designed by first concatenating the traffic data with the external factor data, and then the combined data are input into the KE-GNN (w/o external factors) model. Table 4 shows the results for the three models including "'w/o external factors," concatenation-based model and proposed KE-H-GNN. The best values were highlighted in bold.

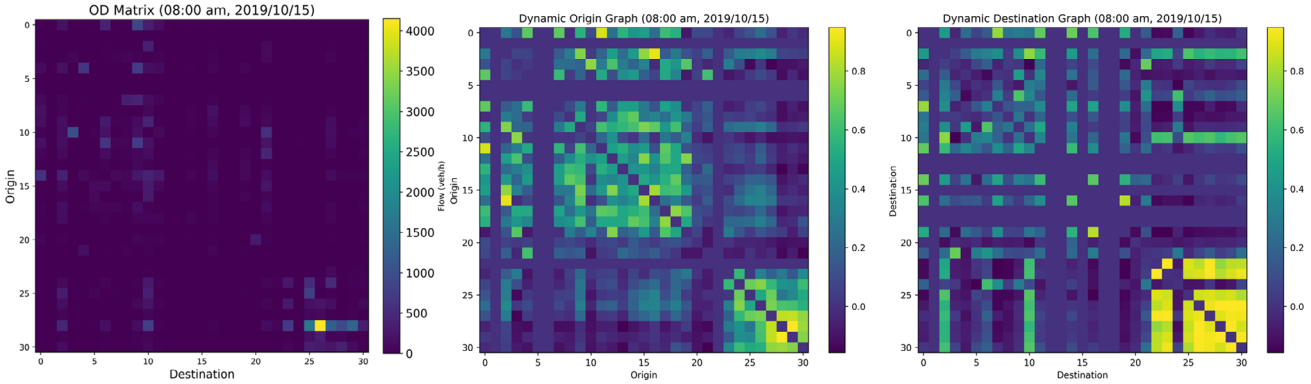
Based on the results, we observe that the KE-H-GNN model outperforms both the models "w/o external factors" and the concatenation-based model. Additionally, we can see that even when considering external factors, the MAPE for the concatenation-based model is slightly higher than that of the "w/o external factors" model in the Tokyo MEX ETC dataset. In the concatenation-based approach, the model treats all external factors equally, regardless of their impact on OD data. This integration method might not sufficiently consider how different external factors interact with each other and with OD data. The primary advantage of using MLR over concatenation is that MLR enables the model to weigh external factors selectively based on their relevance to the OD data, leading to more adaptive prediction. The weights for the external factors are set as learnable parameters within the model through a linear layer, allowing them to be dynamically optimized



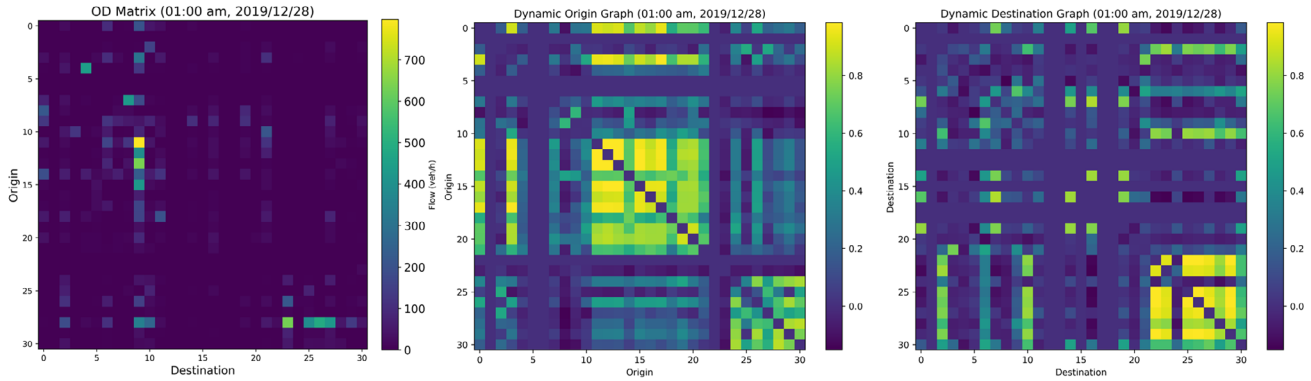
(a) NYC yellow taxi dataset (8:00 a.m., 2023/06/01).



(b) NYC yellow taxi dataset (5:00 a.m., 2023/06/25).



(c) Tokyo MEX ETC dataset (8:00 a.m., 2019/10/15).



(d) Tokyo MEX ETC dataset (1:00 a.m., 2019/12/28).

FIGURE 6 Visualization of OD matrix and dynamic graphs.

TABLE 4 Comparison of concatenation-based model and proposed knowledge-enhanced hybrid spatial–temporal graph neural networks (KE-H-GNN).

Baselines	NYC yellow taxi				Tokyo MEX ETC			
	RMSE (veh/h)	MAE (veh/h)	MAPE (%)	MSSIM	RMSE (veh/h)	MAE (veh/h)	MAPE (%)	MSSIM
W/o external factors	2.88 ± 0.06	0.284 ± 0.01	2.37 ± 0.06	0.9981 ± 0.0002	26.50 ± 0.09	7.32 ± 0.10	13.50 ± 0.13	0.9334 ± 0.0023
Concatenation-based model	2.78 ± 0.11	0.270 ± 0.01	2.36 ± 0.15	0.9980 ± 0.0005	26.01 ± 0.16	7.28 ± 0.05	13.56 ± 0.27	0.9359 ± 0.0003
KE-H-GNN (ours)	**2.70 ± 0.04**	**0.268 ± 0.02**	**2.23 ± 0.11**	**0.9984 ± 0.0003**	**25.92 ± 0.10**	**7.20 ± 0.03**	**13.24 ± 0.24**	**0.9361 ± 0.0004**

TABLE 5 Effects of external factors.

External factors	NYC yellow taxi	Tokyo MEX ETC
Weekend effect	**0.0695 ± 0.0390**	0.1806 ± 0.0608
Holiday effect	0.0664 ± 0.0321	0.1769 ± 0.0712
Rain effect	0.0638 ± 0.0245	**0.1968 ± 0.0660**

during the training process. This approach ensures that influential external variables contribute meaningfully to the prediction, while less relevant factors are naturally downweighed.

A further implication is that the MLR approach to incorporate external data will improve the model's interpretability. To better understand the impact of external factors on ODP, the effect of individual external factor E^i ($E^i \in E$) on OD data M are obtained by calculating a ratio between the output of neural network $f(M)$ to the contribution of the individual external factor $B^i E^i$ during the final training epoch as described in Section 4.4. The results are presented in Table 5. The highest values were highlighted in bold.

From Table 5, we observe that each factor contributes differently to the ODP in both datasets. For NYC yellow taxi data, the weekend effect is the strongest, while the rain effect is the weakest. Interestingly, the rain effect is the strongest for the Tokyo MEX ETC data. Additionally, the combined effects of external factors are more significant in the Tokyo MEX ETC dataset than in the NYC yellow taxi dataset. This observation is consistent with the ablation study results, where the model "w/o external factor" performed better in the NYC yellow taxi dataset. By explicitly weighing each external factor in the regression process, MLR offers deeper insights into which external factors have the most significant impact on ODP, which can be easily understood by practitioners and planners. Therefore, as discussed above, the proposed MLR module for incorporating external factors demonstrates clear advantages over simple concatenation, providing valuable insights for real-world applications.

6.4 | Comparison of the proposed region partitioning strategies

In this part, we compare the proposed two meaningful region partitioning strategies and discuss their potential implications. Table 6 presents the results for both cases. Better values were highlighted in bold.

As shown in Table 6, the election district-based region partition performs better for the Tokyo MEX ETC dataset except for RMSE, which is 1.15 (veh/h) higher. In contrast, for the NYC yellow taxi dataset, it is slightly outperformed by K-means clustering except for RMSE, which is 0.18

TABLE 6 Results for different region partition strategies.

Baselines	Representation	NYC yellow taxi			Tokyo MEX ETC			
		RMSE (veh/h)	MAE (veh/h)	MAPE (%)	MSSIM	RMSE (veh/h)	MAE (veh/h)	MAPE (%)	MSSIM
Election district-based data aggregation	"Mass"	**2.52 ± 0.05**	0.305 ± 0.01	3.34 ± 0.07	0.9968 ± 0.0001	27.07 ± 0.15	**6.87 ± 0.06**	**10.81 ± 0.25**	**0.9435 ± 0.0005**
Clustering-based data aggregation	Distance	2.70 ± 0.04	**0.268 ± 0.02**	**2.23 ± 0.11**	**0.9984 ± 0.0003**	**25.92 ± 0.10**	7.20 ± 0.03	13.24 ± 0.24	0.9361 ± 0.0004

(veh/h) lower. This difference may be attributed to the way NYC taxi service zones are delineated. These predefined taxi zones may have already accounted for factors related to user distribution, which gives K-means clustering an advantage in capturing traffic patterns that are not solely defined by geographic distance. In other words, the pre-defined taxi zones in NYC implicitly considered user distribution, making distance-based clustering better suited to this scenario.

Although K-means clustering-based region partition performed slightly better in the NYC yellow taxi dataset, this approach faces challenges in certain applications. For instance, ETC toll gates are isolated points, and their service areas are not clearly defined. In such cases, the K-means clustering-based approach requires manually defining cluster boundaries (such as with convex hulls), which may not accurately reflect actual traffic zones. This limitation reduces its generalizability to scenarios with distinct geographic points, such as toll gates or subway stations. In addition, K-means clustering relies solely on geographical distance, which may not fully capture complex traffic dynamics within the network. Therefore, incorporating additional data sources, such as point-of-interest data, could provide a more comprehensive framework for clustering, thereby enhancing its generalizability.

In contrast, election district-based region partitioning offers a more straightforward and intuitive approach, making it easier for decision-makers and practitioners to understand and apply in real-world scenarios. However, the election district-based approach may be too large for certain transportation mode planning, particularly for public transportation systems that focus on shorter-distance movements (e.g., people walking to bus stops). This may limit its generalizability in contexts where smaller zones are necessary. To address this, a multi-scale approach may be more effective in accommodating the varying levels of granularity required in different scenarios. For example, in macro-level planning, such as for taxis and expressways, election districts can serve as an effective basis, while micro-level analyses can employ finer-grained partitions, such as functional zones or traffic analysis zones, to ensure precision.

7 | CONCLUSION

In this paper, we introduced KE-H-GNN, a novel ODP model designed to balance prediction performance with interpretability. The model combined a deep learning predictive model incorporating traffic engineering domain knowledge, with an MLR module for integrating external factors.

Building on insights from the gravity model, we proposed two meaningful region partition strategies for aggregating OD matrices: one based on election districts and another using K-means clustering. These aggregated OD matrices, along with static graphs including adjacency and distance matrices, were used as inputs for the prediction model. An LSTM network was first used to capture the temporal relationships in the OD data. To learn spatial relationships, we developed a multi-graph input GCN module consisting of three branches: two branches of GCNs for static graphs and one for dynamic graphs using BDGCN layers. We also introduced a global–local attention module inspired by traffic flow theory to capture comprehensive spatial features within the OD data. Finally, an MLR module was employed to model and quantify the relationship between the predicted OD matrices and external factors.

Experiments conducted on two real-world datasets demonstrated that our proposed framework achieved better prediction performance through effective modeling of the ST relationships in OD data. We found that the election district-based region partitioning approach was more straightforward, compared to K-means clustering-based approach, making it easier for decision-makers to apply in practical contexts. Additionally, we confirmed that the MLR module outperformed traditional concatenation methods for integrating external factors, offering improved prediction performance and interpretability by selectively weighing external factors based on their relevance to the OD data.

In conclusion, the proposed KE-H-GNN not only enhances the ODP accuracy but also provides a deeper understanding of the factors influencing OD demand, making it a useful tool for practitioners who require interpretable models in real-world situations.

Future research will explore the potential for more recent sophisticated classification algorithms to further improve the framework, for example, neural dynamic classification algorithm (Rafiei & Adeli, 2017), dynamic ensemble learning algorithm (Alam et al., 2020), finite element machine (Pereira et al., 2020), and self-supervised learning (Rafiei et al., 2024a, 2024b). Furthermore, while our research focuses on the ODP task, several earlier works provide valuable motivation for extending our methodology to applications in freeway traffic incident detection (Adeli & Karim, 2005; Karim & Adeli, 2003; Samant & Adeli, 2001).

ACKNOWLEDGMENTS

The authors would like to thank Metropolitan Expressway Co. Ltd. for providing the ETC data and the Japan Weather Association for providing the AMEDAS weather data for this research. This work was supported by JST SPRING (Grant Number JPMJSP2108); Key Technologies of Traffic Signal Control Evaluation-Diagnosis-Optimization Driven by Artificial Intelligence and Digital Twin, Innovation and Technology Fund—Mainland-Hong Kong.

Funding information

JST SPRING, Grant Number JPMJSP2108; Key Technologies of Traffic Signal Control Evaluation-Diagnosis-Optimization Driven by Artificial Intelligence and Digital Twin, Innovation and Technology Fund—Mainland-Hong Kong Joint Funding Scheme (ITF-MHKJFS), MHP/038/23

REFERENCES

Adeli, H., & Karim, A. (2005). *Wavelets in intelligent transportation systems*. John Wiley and Sons.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing with Applications*, *32*(10), 8675–8690.

Almeida, T., & Manquinho, V. (2022). Constraint-based electoral districting using a new compactness measure: An application to Portugal. *Computers & Operations Research*, *146*, 105892.

Behara, K. N. S., Bhaskar, A., & Chung, E. (2022). Geographical window based structural similarity index for origin-destination matrices comparison. *Journal of Intelligent Transportation Systems*, *26*(1), 46–67.

Chu, K., Lam, A. Y. S., & Li, V. O. K. (2020). Deep multi-scale convolutional LSTM network for travel demand and origin-destination predictions. *IEEE Transactions on Intelligent Transportation Systems*, *21*(8), 3219–3232.

Chung, E., Ohtani, O., Warita, H., Kuwahara, M., & Morita, H. (2005). Effect of rain on travel demand and traffic accidents. *Proceedings of the 8th International Conference on Intelligent Transportation Systems (ITSC)*, Vienna, Austria (pp. 1080–1083).

Feng, A., & Tassiulas, L. (2022). Adaptive graph spatial-temporal transformer network for traffic forecasting. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22)*, New York, NY (pp. 3933–3937).

Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(1), 922–929.

Han, L., Ma, X., Sun, L., Du, B., Fu, Y., Lv, W., & Xiong, H. (2022). Continuous-time and multi-level graph representation learning for origin-destination demand prediction. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, New York, NY (pp. 516–524).

Hu, J., Yang, B., Guo, C., Jensen, C. S., & Xiong, H. (2020). Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks. *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, Dallas, TX (pp. 1417–1428).

Huang, B., Ruan, K., Yu, W., Xiao, J., Xie, R., & Huang, J. (2023). ODformer: Spatial–temporal transformers for long sequence Origin–Destination matrix forecasting against cross application scenario. *Expert Systems with Applications*, *222*, 119835.

Huang, X., Zhang, B., Feng, S., Ye, Y., & Li, X. (2023). Interpretable local flow attention for multi-step traffic flow prediction. *Neural Networks*, *161*, 25–38.

Jiang, R., Wang, Z., Cai, Z., Yang, C., Fan, Z., Xia, T., Matsubara, G., Mizuseki, H., Song, X., & Shibasaki, R. (2021). Countrywide origin-destination matrix prediction and its application for COVID-19. In Y. Dong, N. Kourtellis, B. Hammer, & J. A. Lozano (Eds.), Lecture notes in computer science: Vol. 12978. *Machine learning and knowledge discovery in databases. Applied data science track. ECML PKDD 2021* (pp. 319–334). Springer.

Karim, A., & Adeli, H. (2003). Fast automatic incident detection on urban and rural freeways using wavelet energy algorithm. *Journal of Transportation Engineering, ASCE*, *129*(1), 57–68.

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*, 102858.

Kipf, T. N., & Welling, M. (2016). *Semi-supervised classification with graph convolutional networks*. https://doi.org/10.48550/arXiv.1609.02907

Li, D., Wang, W., & Zhao, D. (2023). Designing a novel two-stage fusion framework to predict short-term origin–destination flow. *Journal of Transportation Engineering, Part A: Systems*, *149*(5), 04023032.

Li, X., Kurths, J., Gao, C., Zhang, J., Wang, Z., & Zhang, Z. (2017). A hybrid algorithm for estimating origin-destination flows. *IEEE Access*, *6*, 677–687.

Li, Y., Jian, L., Zhang, L., & Zhao, Y. (2017). Taxi booking mobile app order demand prediction based on short-term traffic forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, *2634*, 57–68.

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). *Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. 6th International Conference on Learning Representations (ICLR 2018)*, Vancouver, BC, Canada.

Liu, L., Qiu, Z., Li, G., Wang, Q., Ouyang, W., & Lin, L. (2019). Contextualized Spatial–Temporal Network for Taxi Origin-Destination Demand Prediction. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3875–3887.

Lu, B., Gan, X., Zhang, W., Yao, H., Fu, L., & Wang, X. (2022). Spatio-temporal graph few-shot learning with cross-city knowledge transfer. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, Washington, DC (pp. 1162–1172).

McNally, M. G. (2000). The four step model. *Center for activity systems analysis*. https://escholarship.org/uc/item/7j0003j0

Moreira-Matias, L., Gama, J., Ferreira, M., Moreira, J., & Damas, L. (2013). Predicting taxi-passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, *14*, 1393–1402.

Mukai, N., & Yoden, N. (2012). Taxi demand forecasting based on taxi probe data by neural network. In T. Watanabe, J. Watada, N. Takahashi, R. Howlett, & L. Jain (Eds.), *Intelligent interactive multimedia: Systems and services. smart innovation, systems and technologies* (Vol. 14, pp. 589–597).Springer.

Nishikawa, A. (2022). *Statistical data and map data of the 289 election districts revised in 2022*. (In Japanese). https://gtfs-gis.jp/senkyoku

NYC Districting Commission. (2022). *NYC board of elections 2023 city council election maps*. https://www.nyc.gov/site/districting/maps/maps.page

Peng, D., Huang, M., & Xing, Z. (2023). Taxi origin and destination demand prediction based on deep learning: A review. *Digital Transportation and Safety*, *2*(3), 176–189.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393–6404.

Qi, X., Yao, J., Wang, P., Shi, T., Zhang, Y., & Zhao, X. (2024). Combining weather factors to predict traffic flow: A spatial-temporal fusion graph convolutional network-based deep learning approach. *IET Intelligent Transport Systems*, *18*, 528–539.

Qu, Y., Li, Z., Zhao, X., & Ou, J. (2024). Towards real-world traffic prediction and data imputation: A multi-task pretraining and fine-tuning approach. *Information Sciences*, *657*, 119972.

Qu, Y., Rong, J., Li, Z., & Chen, K. (2023). ST-A-PGCL: Spatiotemporal adaptive periodical graph contrastive learning for traffic prediction under real scenarios. *Knowledge-Based Systems*, *272*, 110591.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Rafiei, M. H., Gauthier, L., Adeli, H., & Takabi, D. (2024a). Self-supervised learning for electroencephalography. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(2), 1457–1471.

Rafiei, M. H., Gauthier, L., Adeli, H., & Takabi, D. (2024b). Self-supervised learning for near-wild cognitive workload estimation. *Journal of Medical Systems*, *48*, 107.

Romanowska, A., & Budzyński, M. (2022). Investigating the impact of weather conditions and time of day on traffic flow characteristics. *Weather, Climate, and Society*, *14*, 823–833.

Samant, A., & Adeli, H. (2001). Enhancing neural network incident detection algorithms using wavelets. *Computer-Aided Civil and Infrastructure Engineering*, *16*(4), 239–245.

Shi, H., Yao, Q., Guo, Q., Li, Y., Zhang, L., Ye, J., Li, Y., & Liu, Y. (2020). Predicting origin-destination flow via multi-perspective graph convolutional network. *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, Dallas, TX (pp. 1818–1821).

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, *30*(3), 83–98.

Tong, Y., Chen, Y., Zhou, Z., Chen, L., Wang, J., Yang, Q., Ye, J., & Lv, W. (2017). The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '23)*, Halifax, NS, Canada (pp. 1653–1662).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, New York, NY (pp. 6000–6010).

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, *43*, 3–19.

Wang, Y., Yin, H., Chen, H., Wo, T., Xu, J., & Zheng, K. (2019). Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling. *Proceedings of the 25th ACM SIGKDD International Conference on Knowl-*

edge Discovery & Data Mining (KDD' 19), Anchorage, AK (pp. 1227–11235).

Xu, H., Chen, Y., Li, C., & Chen, X. (M.) (2024). Space-time adaptive network for origin-destination passenger demand prediction. *Transportation Research Part C: Emerging Technologies*, *167*, 104842.

Xu, J., Rahmatizadeh, R., Bölöni, L., & Turgut, D. (2018). Real-time prediction of taxi demand using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *19*, 2572–2581.

Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., & Xiong, H. (2020). Spatial-Temporal Transformer Networks for Traffic Flow Forecasting. arXiv. https://doi.org/10.48550/ARXIV.2001.02908

Yang, Y., Zhang, S., Zhang, C., & Yu, J. J. Q. (2021). Origin-destination matrix prediction via hexagon-based generated graph. *Proceedings of the 24th IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, IN (pp. 1399–1404).

Yao, H., Liu, Y., Wei, Y., Tang, X., & Li, Z. (2019). Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. *Proceedings of the 2019 World Wide Web Conference (WWW'19)*, San Francisco, CA (pp. 2181–2191).

Yao, X., Gao, Y., Zhu, D., Manley, E., Wang, J., & Liu, Y. (2021). Spatial Origin-Destination Flow Imputation Using Graph Convolutional Networks. *IEEE Transactions on Intelligent Transportation Systems*, *22*(12), 7474–7484.

Ye, Y., Xiao, Y., Zhou, Y., Li, S., Zang, Y., & Zhang, Y. (2023). Dynamic multi-graph neural network for traffic flow prediction incorporating traffic accidents. *Expert Systems with Applications*, *234*, 121101.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 3634–3640.

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*, 102851.

Zhang, J., & Hu, G. (2024). *Origin-destination prediction from road segment speed data using GraphResLSTM model*. Preprints 2024061170. https://www.preprints.org/manuscript/202406.1170/v2

Zhang, R., Han, L., Liu, B., Zeng, J., & Sun, L. (2022). Dynamic graph learning based on hierarchical memory for origin-destination demand prediction. *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI-22)*, Messe Wien, Vienna, Austria (pp. 2383–2389).

Zhang, T., Ding, W., Chen, T., Wang, Z., & Chen, J. (2021). A graph convolutional method for traffic flow prediction in highway network. *Wireless Communications and Mobile Computing*, *2021*, 1997212.

Zhang, X., Liang, X., Zhiyuli, A., Zhang, S., Xu, R., & Wu, B. (2019). AT-LSTM: An attention-based LSTM model for financial time series prediction. *IOP Conference Series: Materials Science and Engineering*, *569*(5), 052037.

Zhang, Y., Xu, S., Zhang, L., Jiang, W., Alam, S., & Xue, D. (2024). Short-term multi-step-ahead sector-based traffic flow prediction based on the attention-enhanced graph convolutional LSTM network (AGC-LSTM). *Neural Computing and Applications*. Advance online publication. https://doi.org/10.1007/s00521-024-09827-3

Zhao, J., Qu, H., Zhao, J., & Jiang, D. (2018). Towards traffic matrix prediction with LSTM recurrent neural networks. *Electronics Letters*, *54*(9), 540–603.

Zhu, L., Yu, F. R., Wang, Y., Ning, B., & Tang, T. (2019). Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *20*(1), 383–398.

Zou, X., Chung, E., Zhou, Y., Long, M., & Lam, W. H. K. (2023). A feature extraction and deep learning approach for network traffic volume prediction considering detector reliability. *Computer-Aided Civil and Infrastructure Engineering*, *39*, 102–119.

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.

How to cite this article: Xing, Z., Chung, E., Wang, Y., Toriumi, A., Oguchi, T., & Wu, Y. (2025). Origin–destination prediction via knowledge-enhanced hybrid learning. *Computer-Aided Civil and Infrastructure Engineering*, *40*, 2498–2521. https://doi.org/10.1111/mice.13458