



AdaTrans: Adaptive Transfer Time Prediction for Multi-modal Transportation Modes

Shuxin Zhong
Rutgers University
Piscataway, NJ, USA
shuxin.zhong@rutgers.edu

Hua Wei
Arizona State University
Tempe, AZ, USA
hua.wei@asu.edu

Wenjun Lyu
Rutgers University
Piscataway, NJ, USA
wenjun.lyu@rutgers.edu

Guang Yang
Rutgers University
Piscataway, NJ, USA
gy121@rutgers.edu

Zhiqing Hong
Rutgers University
Piscataway, NJ, USA
zhiqing.hong@rutgers.edu

Guang Wang
Florida State University
Tallahassee, FL, USA
guang@fsu.edu

Yu Yang
Lehigh University
Bethlehem, PA, USA
yuyang@lehigh.edu

Desheng Zhang
Rutgers University
Piscataway, NJ, USA
desheng@cs.rutgers.edu

Abstract

Multi-modal transportation leverages the advantages of various transportation modes, leading to more efficient urban traveling services. Accurately predicting transfer times between different modes provides guidance for tasks such as trip planning and transportation management. Most existing transfer time prediction works rely on strong assumptions, e.g., predetermined routes, assumed speeds, and predefined downstream transportation timetables. However, these assumptions are hard to hold in practice due to *internal factors* like individual preferences and *external factors* like dynamic traffic conditions. These factors are dynamic and vary with location and time, presenting a significant challenge. To address this, we introduce an adaptive transfer time prediction framework, AdaTrans, to forecast personalized transfer times between upstream and downstream transportation modes. Firstly, an attribute learning module is designed to model the trends of internal factors. Then a spatial-temporal adaptive learning component is designed to learn dynamic external factors. Finally, an aggregation component with a capsule network is employed to fuse the influences of these factors. The extensive evaluation results in two real-world datasets demonstrate that AdaTrans effectively harnesses insights from internal and external factors, outperforming state-of-the-art methods by $\sim 20\%$.

CCS Concepts

• Information systems → Data mining.

Keywords

Multi-modal Transportation Modes, Transfer Time, Spatial-temporal Prediction

ACM Reference Format:

Shuxin Zhong, Hua Wei, Wenjun Lyu, Guang Yang, Zhiqing Hong, Guang Wang, Yu Yang, and Desheng Zhang. 2024. AdaTrans: Adaptive Transfer Time Prediction for Multi-modal Transportation Modes. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*, October 21–25, 2024, Boise, ID, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3627673.3679585>

1 Introduction

Multi-modal transportation, e.g., cycling, driving, and public transit, facilitates urban mobility [28, 31, 31, 40, 44, 45]. For example, as shown in Figure 1, a citizen might take the subway from the origin to the nearest station for rapid transit, then switch to a bus to reach the final destination if the subway does not directly serve it. The time taken to transfer from the subway to the bus, referred to as *Transfer Time*, is a key component of the total travel time. According to the Transit Cooperative Research Program [8], transfer time accounts for over 25% of the total travel time on average. Therefore, accurately predicting transfer time is crucial for enhancing passenger experience and improving the management of urban transportation systems [21].

Current studies [24, 25, 27] on multi-modal transportation transfer time prediction generally assume that passengers follow recommended routes and walk at an average speed to board the next available mode of transportation. Consequently, transfer times are derived from upstream and downstream transportation timetables, resulting in identical predicted transfer times for passengers with the same origin and destination station pairs. As illustrated in Figure 2 (a), this approach underestimates transfer times, with over 80% of the estimated times being less than 20 minutes, while in reality, over 50% of the actual times exceed 25 minutes. Furthermore, as shown in Figure 2 (b), we analyze transfer times for two station pairs, one in an urban area and the other in a suburban area. The

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '24, October 21–25, 2024, Boise, ID, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0436-9/24/10

<https://doi.org/10.1145/3627673.3679585>

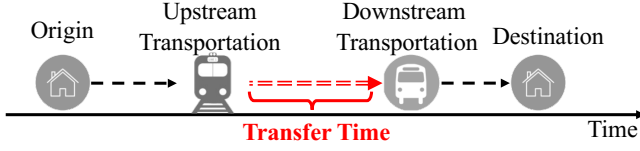


Figure 1: Transferring in multi-modal transportation.

analysis compares transfer times during rush hours and non-rush hours, revealing significant variation among different passengers. There are distinct differences in transfer times not only between the two areas but also across different time periods.

The variation in transfer time is primarily due to two factors: i) *internal factors*, stemming from the intrinsic unpredictability of passengers. For example, passengers may arrive at the same bus station via different walkways [5], which may not align with the recommended route shown on the navigation software. Additionally, passengers may have different walking speeds based on personal preferences and behaviors [29]. ii) *external factors*, arising from the specific characteristics of downstream transportation and congestion conditions. For example, when a bus is overcrowded, some passengers may have to wait for subsequent buses, resulting in an extended transfer time [16, 41].

These spatial-temporal variabilities pose a significant challenge. Therefore, we propose AdaTrans, an Adaptive Transfer time prediction framework for personalized transfer times between upstream and downstream transportation modes. AdaTrans consists of three components: i) an attribute learning component to capture the trends of *internal factors*, ii) a spatial-temporal adaptive learning component to model the *external factors*, and iii) an adaptive aggregation module to combine these factors for more accurate personalized transfer time prediction. In summary, this paper makes the following three contributions:

- To the best of our knowledge, we are the first to design an adaptive transfer time prediction framework, named AdaTrans, for personalized transfer time prediction in multi-modal transportation scenarios. Unlike previous works that primarily consider real-time traffic, AdaTrans effectively models dynamic influences from internal factors (e.g., passengers' preferences) and external factors (e.g., the latest demand and supply).
- We have two major technical designs: i) a spatial-temporal adaptive learning component that models the multi-modal transportation stations as a heterogeneous graph and adaptively learns the spatial-temporal correlations between stations, and ii) an attribute learning component that leverages personalized preferences by considering the similarity of travelers' patterns.
- We conduct comprehensive experiments using two real-world datasets from Shenzhen, China. The results show that AdaTrans outperforms state-of-the-art methods by approximately 20% on average. Moreover, we conducted detailed ablation studies and sensitivity analyses.

The rest of the paper is organized as follows. In Section 5, we discuss the related work. In Section 2, we provide the formal definition of the transfer time prediction problem. We show the detailed

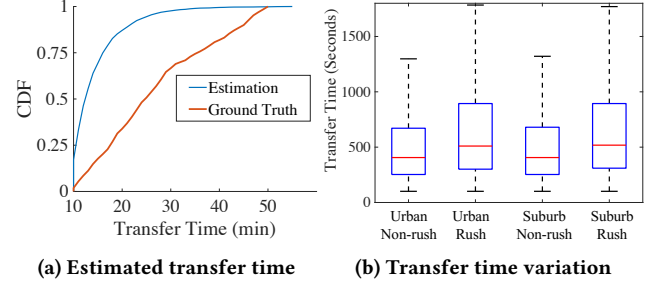


Figure 2: Motivations for transfer time prediction. (a) Transfer time estimated by timetables compared to ground truth; (b) Transfer time varied by location pairs and periods.

design in Section 3, including the framework overview and technical explanation. In Section 4, we evaluate the performance of AdaTrans to show the improvement compared with SOTA and conduct detailed ablation studies and parameter sensitivity analyses. Finally, we conclude the paper in Section 6.

2 Preliminaries

Given a query q , including passenger id p_i , time t_{u_i} arriving at upstream station u_i , and a downstream transportation mode d_j , we aim at predicting the transfer time $t_{(u_i, d_j)}$. It represents the interval between the departure from the upstream station and the arrival at the downstream transportation mode. Formally, it is expressed as:

$$t_{(u_i, d_j)} = \mathcal{F}_\theta(q) = \mathcal{F}_\theta(p_i, t_{u_i}, u_i, d_j) \quad (1)$$

where θ denotes all learnable parameters. In practical usage, d_j is either estimated by prediction methods or manual inputs from passengers, which is beyond the scope of this paper.

Further, to model the time-varying spatial relationships between upstream stations and downstream transportation, we formulate it on dynamic heterogeneous graph using Eq. 2.

$$t_{(u_i, d_j)} = \mathcal{F}_\theta(q; \mathcal{G}_t) = \mathcal{F}_\theta(p_i, u_i, d_j, t_{u_i}; \mathcal{G}_t) \quad (2)$$

Here, $\mathcal{G}_t = (\mathcal{V}, \mathcal{A}_t)$ is a *station-transportation mode* graph at t , where $\mathcal{V} = \mathcal{V}_u \cup \mathcal{V}_d$ is the set of heterogeneous nodes, $\mathcal{V}_u$ and $\mathcal{V}_d$ indicate the upstream stations and downstream transportation, respectively. $\mathcal{A}_t$ represents the connection features (e.g., reachability) from upstream stations to downstream transportation at t . It is worth mentioning that our definition also works for transfer time between the same transportation mode (e.g., from one subway line to another) depending on whether the $\mathcal{V}$ is heterogeneous or homogeneous. Without loss of generality, we discuss the problem based on the setting of heterogeneous nodes.

3 Methodology

Figure 3 illustrates the architecture of AdaTrans, a framework designed to predict personalized transfer times for trip records based on historical transfer-flow data and trip query attributes. It is divided into three main components:

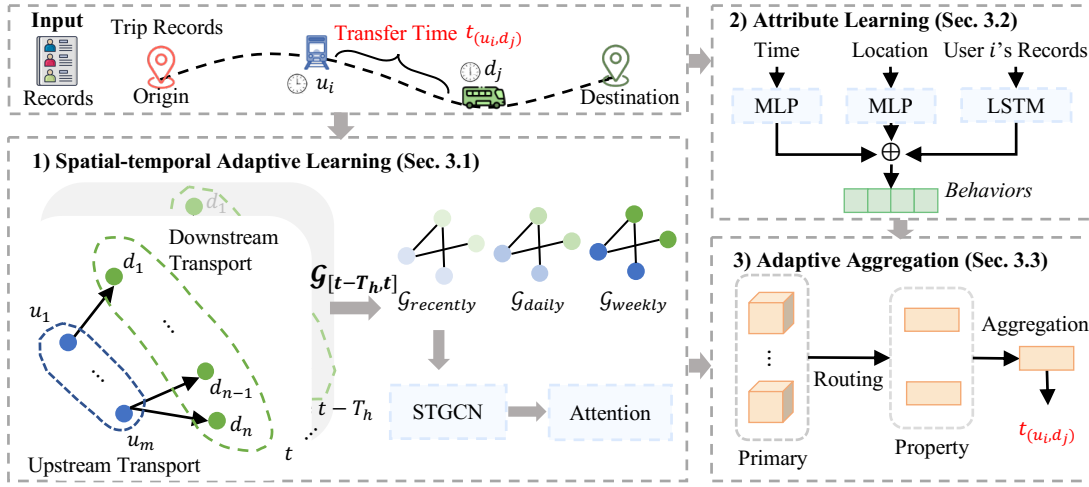


Figure 3: The framework of AdaTrans. It begins with the spatial-temporal adaptive learning component, where the input includes node features (e.g., time and the number of passengers entering and departing the transportation mode) and edge features (e.g., time and average transfer time). This component captures the influence of *external* factors (detailed in Section 3.1) while also recognizing recent, daily-periodic, and weekly-periodic correlations in the data. Next, the attribute learning component processes inputs related to a query q , containing personalized features (e.g., passenger ID, time, and location). This component models the impact of *internal* factors (elaborated in Section 3.2). The derived features are then fed into the aggregation component, which includes primary capsules, property capsules, and aggregation capsules (explained in Section 3.3). The output is then passed through a fully connected layer to produce the predicted transfer time $t_{(u_i, d_j)}$.

- *Spatial-Temporal Adaptive Learning*: This component is tasked with encoding *external* factors. Specifically, it captures time-dependent spatial correlations between different transportation modes. Within this component, there are three sub-modules, each having identical structures, to address recent, daily-periodic, and weekly-periodic patterns in the input data [15].
- *Attribute Learning*: Focusing on *internal* factors, it understands the influence of individual passenger habits on transfer times.
- *Aggregation*: This step fuses the extracted factors, creating a comprehensive representation for predicting transfer times.

3.1 Spatial-temporal Adaptive Learning

Transfer times between upstream and downstream transportation stations are influenced both by geographical factors (e.g., distance and location) and by dynamic real-time factors (e.g., demand and supply). For instance, stations situated in commercial zones typically see a surge in demand for downstream transportation during morning peak hours, aligning with the commuter influx. Conversely, residential areas witness the opposite trend. The imbalance between demand and supply, especially evident during rush hours in business districts, can lead to extended wait times owing to the disproportionate ratio of passengers to available seating. Recognizing the need to address these time-sensitive external factor influences, we introduce a spatial-temporal adaptive learning component.

The input is historical transfer-flow features denoted as $\mathcal{X} = \{X_1, \dots, X_t\} \in \mathbb{R}^{N_t \times N_u \times N_d \times |f_{in}|}$. N_t , N_u , and N_d correspond to the number of past time slots, upstream stations, and downstream

transportation modes, respectively. The input comprises features of dimension $|f_{in}|$, which encapsulate the embeddings associated with p_i , u_i , and d_i . Our approach involves a two-pronged process: first, we construct a heterogeneous adaptive graph, and subsequently, we engage in learning the correlations. The objective is to determine the general transfer-flow feature for the subsequent time slot $t + 1$.

3.1.1 Heterogeneous adaptive graph construction. Current adaptive graph learning frameworks [3, 14, 42], predominantly target node features (i.e., region-specific travel time predictions). In contrast, our emphasis is on edge features, specifically the transfer time between two nodes. To adapt these existing frameworks to our context, two potential approaches arise: i) utilizing the features of the two nodes that constitute an edge to represent that edge, and ii) considering the edges themselves as nodes. Despite these adaptation strategies, two primary challenges persist. The first challenge is that these existing frameworks are constructed upon homogeneous graphs, which might not adeptly distinguish between the unique characteristics of upstream stations and their downstream transportation counterparts. The second limitation is the static nature of the adjacency matrix A , which does not account for the dynamic, time-sensitive nuances of spatial relationships. As an illustration, bus routes might be varied considering passenger needs, accommodating different schedules (e.g., holidays, nighttime [1]).

To address the aforementioned challenges, we introduce the time-specific bi-adjacent matrix, denoted as $B \in \mathbb{R}^{N_u \times N_d}$, which symbolizes the heterogeneous transfer graph. Formally, this can be presented as: $A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}$. Also, we denote the degree matrix

by $D = \begin{bmatrix} D_u & 0 \\ 0 & D_d \end{bmatrix}$, where D_u and D_d are the degree matrix of the upstream stations and downstream transportation. For effective flow feature prediction, we conceptualize the transfer flow as nodes in a graph, linked by stations. This transfer graph undergoes a transformation into its line graph, as indicated by [2]. This conversion can be mathematically captured by:

$$L(B)_{ijkl} = \begin{cases} B_{ij} \cdot B_{kl} & \text{if } i = k \text{ or } j = l \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where i, j, k, l serve as dimensional indices, with i and j characterizing the first dimension k and l detailing the second. Conceptually, $L(B)$ can be viewed as the sum of the adjacency matrix and the identity matrix of the line graph, given by

$$L(B) = \tilde{A}_L = I_L + A_L \quad (4)$$

where A_L represents the adjacency matrix associated with the line graph, and I_L is its respective identity matrix.

To reduce computational costs, inspired by [18], the graph convolution operation is adeptly approximated as:

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} XW + b \quad (5)$$

where W and b are learnable weights and biases, $\tilde{A}$ denotes the adjacency matrix augmented by the identity matrix, and $\tilde{D}$ is the degree matrix enhanced by the identity matrix. Given the inherent heterogeneity, we further refine the convolution operation using $L(B)$:

$$Z = \tilde{D}^{-\frac{1}{2}} L(B) \tilde{D}^{-\frac{1}{2}} XW + b \quad (6)$$

Building on the renormalization trick from [18], we substitute $\tilde{D}^{-\frac{1}{2}} L(B) \tilde{D}^{-\frac{1}{2}}$ with $L(D_u^{-1}B)$ because of the non-symmetric matrix B . By doing so, we successfully adapt the GCN to accommodate the bi-adjacency matrix alongside flow inputs, resulting in:

$$\tilde{Z} = L(D_u^{-1}B)XW + b \quad (7)$$

This extended formulation allows for a more robust and nuanced capture of features while ensuring computational efficiency.

Given the dynamic nature of connections between upstream and downstream transportation [1], which affects passenger preferences, we generate B^t adaptively from the inputs rather than relying on a static, predefined bi-adjacency matrix B . Inspired by [3], we employ decomposition as a strategy to approximate the spatial correlations between upstream stations and downstream transportation. Specifically, the formulation is given by:

$$D_u^{-1}B^t = \text{softmax}(\text{ReLU}(E_u^t E_d^{tT})) \quad (8)$$

The $E_u^t \in \mathbb{R}^{N_u \times f}$ and $E_d^t \in \mathbb{R}^{N_d \times f}$ are node embeddings for the upstream station and downstream transportation, respectively, where f is the embedding dimension.

3.1.2 Spatio-temporal correlation learning. To further adapt to the time-specific spatial correlations, we can apply a high-dimension learnable parameter $\Theta \in \mathbb{R}^{N_t \times N_d \times N_u \times f}$ by expanding Eq. 8 as:

$$\Theta = \text{softmax}(\text{ReLU}(E_u E_d^T) \otimes E_t) \quad (9)$$

$E_t \in \mathbb{R}^{N_t \times f}$ is the time-slot embedding, and $\otimes$ is the outer product. The computational cost of Eq. 9 is $O(N_t \cdot N_d \cdot N_u)$, which presents a challenge when dealing with expansive transportation systems,

given its high resource demands [3]. To mitigate this challenge, we propose an efficient partial updating strategy.

For the given input $X \in \mathbb{R}^{|B| \times f_{in}}$, where $|B|$ symbolizes the batch size, we partition it to devise three distinct masking matrices using one-hot vectors. These are: i) Downstream matrix C_d represents downstream transportation; ii) Upstream matrix C_u denotes upstream transportation; and iii) Temporal matrix C_t marks different time slots. Within C_d , C_u , and C_t , each individual row stands as a one-hot encoded vector that signifies the index of the upstream station, downstream station, and temporal instance, respectively. Leveraging these matrices, we are able to extract a more compact feature vector for each training iteration. Specifically, by deploying the operation $C_u \times C_d \times C_t \times \Theta$, only a subset of parameters related to the input is selected. This judicious parameter selection results in a substantial reduction in computational overhead, thus making the process more scalable and efficient.

For each time slot t , the matrix Θ_t encapsulates the specific spatial features between upstream and downstream transportation. We employ the STGCN [42] to discern spatial-temporal relationships across time slots and between stations. Using a sequence of spatial-temporal features $\Theta_{(t-h:t-1)}$ as input, the relationships between nodes are predicted by:

$$H = L(\Theta)X E_u E_d W + E_u E_d b \quad (10)$$

where $L(\cdot)$ transforms Θ into the line graph, $W \in \mathbb{R}^{N_t \times N_u \times N_d \times f_{out}}$, and $b \in \mathbb{R}^{f \times f_{out}}$. The outcome, H_t represents link features, which are derived by adjusting H with Θ_t .

Upon extracting the spatial-temporal graph, we segment it into three periods: recent, daily, and weekly. Each segment is inputted to the STGCN [42] to discern geographical correlations, such as traffic patterns of neighboring stations. The outputs from these segments are then passed through an attention layer, which aggregates the features by weighing their significance. This approach prioritizes the most relevant time-based features for predictions.

3.2 Attribute Learning

Apart from the *external* features (e.g., traffic conditions), the *internal* features (e.g., personalized behaviors) significantly impact transfer time. This influence stems from two main reasons: 1) Passengers often exhibit consistent behaviors, like choosing specific exits, preferring certain routes, and maintaining a relatively consistent speed [29]. 2) They might adhere to particular behavioral patterns, such as traveling during rush and non-rush hours. Given these insights, we have devised an attribute learning approach for understanding time-specific passengers' preferences, which we subsequently employ for personalized transfer time predictions.

A straightforward approach involves learning a personal matrix, $\Omega \in \mathbb{R}^{N_t \times N_p \times f}$, to characterize the behaviors of each passenger. Here, N_t designates the time slots, N_p signifies the number of passengers, and f represents the feature dimensions. Initialized through random encoding, Ω undergoes updates via back-propagation. However, this method presents two notable constraints: 1) As the number of passengers increases, the computational burden significantly increases, making the approach less efficient. 2)

Although the time dimension provides a means to depict varied behaviors of passengers over different spans, the resulting embedding remains static, unable to adapt to changing user preferences.

To address the aforementioned limitations, we introduce an embedding-sharing layer coupled with a recurrent preference embedding technique to enable user-specific attribute learning. This system takes as input the embeddings of the passenger ID, current time, upstream station, and historical trip records, aiming to predict the specific behaviors of a user at a designated time and location. The following is a step-by-step breakdown of our methodology:

- We initialize a two-dimensional matrix, $\Omega \in \mathbb{R}^{N_c \times f}$, to capture the characteristics of representative users. In this matrix, each vector symbolizes a distinct type of user, with N_c being significantly smaller than N_p .
- The embedding-sharing layer translates a user into a vector. From this, we ascertain the most similar representative user preference, p_i , from Ω based on cosine similarity, denoted as ω_{p_i} .
- To predict a passenger's behavior, we execute an addition between the embedding of the most recent time and location with the extracted user preference, ω_{p_i} .

This approach not only optimizes computational efficiency but also ensures dynamic adaptation to the evolving preferences of users. At last, the predicted behavior is integrated into an MLP-based component, tailored for individualized preference learning. This behavior is comprehensively trained throughout the system. Upon completion of the training, the refreshed representation is manually fed back into Ω_{p_i} , serving as the most recent record of user preference.

3.3 Adaptive Aggregation

Once the embeddings of external and internal factors are acquired, we introduce an aggregation component, which organically integrates the aforementioned factors, yielding a time-specific representation of a passenger's transfer time.

Drawing inspiration from the success of the capsule network in computer vision [30], which effectively channels lower-level features like eyes and mouth to corresponding upper-level features such as faces. We integrate it into our aggregation component based on the following insights: 1) Each capsule comprises a group of neurons, representing multifaceted traffic properties in our model, such as traffic volume, supply, and demand, for a more detailed feature representation. 2) The dynamic routing mechanism discerns the significance of local properties to a broader objective (e.g., predicting transfer time). It then increases the coupling coefficient of the properties that are correlated, while diminishing others.

Informed by these insights, we have developed a three-layered capsule network adept at extracting and aggregating information from both external and internal factors. As depicted in Figure 3, the primary capsule ingests the most recent route features and personal attributes, subsequently producing the trip status. This relationship can be mathematically articulated as:

$$S = H_t \oplus \omega_{p_i} \quad (11)$$

Subsequent to this, the dynamic routing mechanism comes into play, isolating pertinent properties as denoted in Eq. 12:

$$v_j = \sum_i W_{ij} s_i \cdot c_{ij} \quad (12)$$

Here, $s_i \in S$ represents the trip status, encapsulating both the link features and user preference. v_j , on the other hand, signifies the properties of the trip when taking into account both the passenger and the prevailing traffic data. Meanwhile, c_{ij} stands as the coupling coefficient, its value being dictated by the dynamic routing procedure. More specifically, the property capsules focus on extracting features pertinent to a range of trip properties, encompassing various travel objectives like commuting, OD pairs, leisure, and casual trips [34]. The subsequent dynamic routing layer allows us to abstract a more generalized representation of transfer time by discerning the contributions of distinct trip patterns.

3.3.1 Further explanation of choosing capsule network. To better comprehend the merits of employing capsule networks for aggregation, we examine other prevalent aggregation techniques, such as concatenation, weighting, and attention mechanisms.

Concatenation-based aggregation: A basic approach involves concatenating the internal and external factors for prediction. Formally, this can be represented as:

$$Z = f(H_t) + \omega_{p_i} + b \quad (13)$$

Here, $+$ denotes summation, and $f(\cdot)$ is a fully connected layer responsible for mapping the edge embedding to the external factor embedding. However, this method comes with an inherent limitation: it is unable to discern and weigh the significance of the two factors, often resulting in sub-performance.

Weighting-based aggregation: To address this limitation, we apply a weighting mechanism that introduces trainable weights for both factors, which is expressed as:

$$Z = \alpha_1 \times f(H_t) + \alpha_2 \times \omega_{p_i} + b \quad (14)$$

where $+$ indicates summation. Both α_1 and α_2 are intrinsically tied to the inputs and are fine-tuned through back-propagation. While this method mitigates the challenge of uniform weight allocation seen in the previous approach [19], it struggles to effectively discern the variances between passengers shaped by the correlation of external and internal factors.

Attention-based aggregation: To address this issue, we integrate an attention mechanism that quantifies the correlation between the factors and the transfer time. It allocates varying levels of attention to the characteristics of passengers as expressed by:

$$Z = a(f(H_t)) \times f(H_t) + a(\omega_{p_i}) \times \omega_{p_i} + b \quad (15)$$

Here, a denotes the attention mechanism and $+$ represents summation. It intuitively models the relationships between transfer time and the two factors by giving prominence to more "correlated" factors while diminishing the weight of less relevant ones.

Summary: While both the attention mechanism and the capsule network excel in modeling the contributions of influential factors to the final output, the capsule network holds an edge in performance. This superiority stems from the neurons within the capsule, which

adeptly model diverse traffic properties, enabling a more granular feature representation [30].

3.4 Training Objective

We utilize the L1 loss as the training objective, and consider the task of predicting the transfer time. Its loss function is given by:

$$\mathcal{L}(\mathbf{W}_\theta) = \sum_{k=1}^N |t_{(u_i, d_j)} - \hat{t}_{(u_i, d_j)}| \quad (16)$$

Here, $\mathbf{W}_\theta$ stands for the set of learnable parameters. $t_{(u_i, d_j)}$ and $\hat{t}_{(u_i, d_j)}$ denote the actual and predicted transfer time, respectively.

4 Experiments

In this section, we introduce the datasets, evaluation metrics, and baselines. Then, we represent the implementation details and experiment settings. At last, we evaluate the performance of AdaTrans compared with different baselines and conduct an ablation study and sensitivity analysis.

4.1 Experimental Settings

4.1.1 Dataset description. We collected the public transportation tap-in and tap-out data in June 2018 and September 2019 from Shenzhen City. It involved 3.6 million passengers, 112 subway stations, and 858 bus lines. The dataset differentiates between subway trip records, which capture both entry and exit details, and bus trip records which only log boarding information. Notably, in Shenzhen's short-distance bus routes, like in cities such as New York and Beijing, passengers are charged a flat rate and don't tap out upon exiting. However, since both bus and subway rides utilize the same transportation card, it is feasible to analyze transfer behaviors. This data, being updated in real-time from ticketing systems, offers a live view of vehicular and passenger statuses.

4.1.2 Implementation. The experimental models were constructed using Keras 2.4 and executed on a server powered by an NVIDIA A4000 GPU and Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz with 256GB RAM. For dataset partitioning, transfer records from the initial 10 days were designated as the training set, the subsequent 10 days for validation, and the concluding 10 days for testing. Adam optimization algorithm was selected for training, with a learning rate and decay rate of 0.01 and 0.005 respectively. The training spanned 50 epochs, with model optimization via 5-fold cross-validation. Embedding dimensions for the time slot, day, and demand/supply factors were 6, 3, and 4 respectively, while personalized characteristics and spatial-temporal features both had a dimension of 8.

4.1.3 Baselines. To evaluate the performance of AdaTrans, we compare it with the following baselines. We primarily chose the baselines that target travel time prediction. We also remove the attribute learning as AdaTrans-, to eliminate the personalized characteristics. For instance, DeepTTE and HetETA consider transfer time from an individual's perspective, while STGCN and AGCRN concentrate on a station-level analysis.

- **Historical Average (HA):** We calculate the average transfer time for each upstream and downstream station pair at different time slots, e.g., 8:00-9:00 AM.
- **GBDT [11]:** Gradient Boosting Decision Tree is a widely used ensemble method. The input of GBDT is exactly the same as the input of AdaTrans including all the outputs from the attribute component and the normalized trip records.
- **DeepTTE [32]:** This is an end-to-end framework that estimates the travel times of individual road segments and then sums up these times. Our problem can be simplified to predict the "travel time" given the upstream and downstream stations. Thus, the input for the spatial-temporal learning component in DeepTTE is the GPS points for stations, and the input for the attribute component is userID, time, and distance. We remove the multi-task component to simplify it for one road segment prediction.
- **HetETA [17]:** We simplify the multi-relational graph to a single-relational graph, where the edges denote the reachability between two stations. The features of nodes contain static features (e.g., distance) and dynamic features (e.g., transfer time, passenger volume) regarding time. The departure query in HetETA is set as $q = (u_i, o_i, t_i)$, where passenger u_i leaves the upstream station o_i at time t_i . Our goal is to estimate the transfer time y_i after boarding at the downstream station d_i .
- **STGCN [42]:** The nodes represent both upstream and downstream stations. We compute the adjacency matrix based on the Euclidean distance. To predict the transfer time between a specific pair of stations, such as stations A and B, we employ the STGCN model to incorporate the influence of surrounding station pairs, such as A-C and A-D, for a more accurate transfer time prediction between A and B.
- **AGCRN [3]:** The input is similar to AdaTrans, allowing it to learn adaptive relationships. The output represents the transfer time features of station pairs, which are transformed into predictions. The differences between AGCRN and AdaTrans are: i) AGCRN is designed for symmetric graphs, and not suitable for the directional correlations between stations; ii) the learned adjacent matrix remains static, which cannot model the varying transfer times, e.g., during rush and non-rush hours.

We use Mean Absolute Percentage Error (**MAPE**), Mean Average Error (**MAE**), and Root Mean Squared Error (**RMSE**), to evaluate the performance of the above models.

4.2 Performance Comparison

In Table 1, we present performance metrics in the format "mean ± standard deviation". Relying solely on the HA, which is the mean of past transfer times, gives less accurate predictions, as evidenced by its highest scores across all metrics. GBDT, which uses decision trees and boosting techniques, outperforms the HA approach. Notably, both HA and GBDT consistently yield similar outcomes when tested on the same dataset. The reason behind this consistency is twofold: HA simply averages historical data without introducing any randomness, and our GBDT model operates deterministically.

DeepTTE, by integrating spatial and temporal aspects, demonstrates superior performance over GBDT, reflecting lower MAPE,

Table 1: Performance comparison of AdaTrans with baseline methods on two datasets w.r.t RMSE, MAE, and MAPE with standard deviations. The lower, the better. The best results are bolded and the second-best results are underlined. AdaTrans achieves the best performances.

Model	Dataset	Shenzhen Dataset (2018/6)			Shenzhen Dataset (2019/9)		
	Metrics	RMSE	MAE (<i>min</i>)	MAPE (%)	RMSE	MAE (<i>min</i>)	MAPE (%)
HA		20.69	15.72	62.8	16.49	12.53	45.8
GBDT		14.48	9.90	54.4	10.67	7.53	41.2
DeepTTE		3.04±0.21	2.57±0.17	29.7±1.8	3.18±0.25	2.37±0.12	30.2±1.4
HetETA		5.17±0.29	4.11±0.16	35.3±2.2	3.52±0.15	3.37±0.15	34.8±1.2
STGCN		12.74±0.28	6.68±0.11	45.3±1.4	11.89±0.16	7.95±0.17	44.8±1.8
ARGCN		8.58±0.25	4.72±0.19	38.3±1.7	8.74±0.29	5.09±0.18	40.6±1.3
AdaTrans-		<u>2.66±0.25</u>	<u>1.90± 0.21</u>	<u>24.8 ± 1.9</u>	<u>2.73±0.15</u>	<u>2.01±0.18</u>	<u>26.5 ± 1.2</u>
AdaTrans		2.27±0.23	1.48± 0.23	23.6 ± 2.4	2.49±0.23	1.73±0.28	25.4 ± 1.6

MAE, and RMSE values. This underscores the merit of embedding spatial-temporal correlations into graph data. Yet, the shared parameters of DeepTTE do not fully capture traffic dynamics. HetETA, on the other hand, underperforms compared to DeepTTE, primarily because it generalizes all relationships in its road map, limiting its ability to exploit the intricacies of a multi-relational network.

Diving deeper, we assessed the potential of STGCN and AGCRN models at the station level against our model. The reliance of STGCN on a fixed graph structure, defined by distance metrics, restricts its capability to understand the nuanced interplay between upstream stations and downstream travel. AGCRN overcomes this constraint by adopting adaptive graph learning, achieving a notable RMSE improvement to 8.58 and 8.74 across the two datasets. This enhancement underlines the pivotal role of adaptive learning in understanding transportation dynamics. Our proposed model excels further than AGCRN by recognizing time-specific spatial correlations and factoring in passengers' preferences, resulting in a more refined performance.

4.3 Ablation Study

We conducted ablation studies to assess the individual contributions of different components in our model, specifically focusing on internal, external, and spatial-temporal factors. During each test, we omitted one factor while retaining the rest (detailed in Table 2).

- **Internal Factors:** By evaluating the impact of internal factors, we integrated normalized passenger data with the spatial-temporal component's output, channeling the combined data directly to the aggregation stage. This configuration yielded RMSE and MAE scores of approximately 4.84 and 2.46, respectively. When contrasted with our full model's performance (i.e., 2.27 and 1.48), it is evident that different passengers have unique behaviors. Capturing these behaviors leads to more precise personalized transfer time predictions.
- **External Factors:** To understand the influence of external factors, we excluded real-time data inputs, such as time of day and day of the week. This exclusion resulted in a slight

Table 2: Performance comparison of ablation studies by removing different factors (i.e., external factors, spatial-temporal (ST) factors, and internal factors).

	RMSE	MAE (<i>min</i>)	MAPE (%)
w/o external	<u>2.66±0.27</u>	<u>1.90±0.23</u>	33.3±2.4
w/o ST	3.47±0.24	2.08±0.21	<u>28.6±2.1</u>
w/o internal	4.84±0.31	2.46±0.25	33.1±1.9
AdaTrans	2.27±0.23	1.48±0.23	23.6±2.4

performance dip, with the RMSE value rising to about 2.66 from the previous 2.28. This suggests that contextual data, like time and weekday, subtly affect transfer time prediction accuracy. As expected, peak times like rush hours, characterized by increased demand, tend to have more frequent vehicle scheduling.

- **Spatial-Temporal Factors:** Upon omitting recent traffic variables, such as counts of inbound and outbound passengers, we observed a stark deterioration in model performance, registering a 27.66% decline. This decline resonates with the inherently spatial-temporal nature of traffic conditions.

4.4 Effects of Different Aggregation Methods

To assess the effectiveness of our aggregation component, we designed three variants:

- **Concatenation:** This basic version uses concatenation for aggregation. It employs a 3-layer MLP to process output features, ensuring its model depth is comparable to the other variants.
- **Weighting:** It employs a weighting method for aggregation, thereby emphasizing the significance of each factor differently.

Table 3: Performance comparison of different aggregation methods. The capsule-based aggregation applied in AdaTrans achieves the best performance.

	RMSE	MAE (min)	MAPE (%)
Concatenation	12.32±1.19	6.22±0.47	60.3±4.2
Weighting	6.56±0.52	2.92±0.31	33.8±3.1
Attention	4.01±0.33	2.35±0.35	29.6±2.2
Capsule	2.27±0.23	1.48±0.23	23.6±2.4

- Attention: It integrates an attention mechanism for aggregation. Here, the context is derived from the output of various components, while the query is based on the input trip record.

From our findings presented in Table 3, a few key insights emerged:

- Weighting v.s. Concatenation: The Weighting outperformed Concatenation. It underscores the value of attributing unique weights to individual components rather than combining them.
- Attention v.s. Weighting: The Attention surpassed Weighting in performance. It is attributed that Weighting assigns a uniform weight across all inputs, thus neglecting to consider variations in passengers' responses to internal and external factors.
- The superiority of AdaTrans: The AdaTrans enhanced the performance of Attention. This is due to two primary reasons: i) its vector representation holistically encapsulates multiple attributes of each factor, and ii) the dynamic routing bolsters the categorization of passengers based on their travel behaviors.

4.5 Sensitivity Analysis

We conduct experiments to evaluate the influence of the length of historical time steps in the capsule networks and the embedding dimension in factors learning (i.e., external and internal).

4.5.1 Length of historical time steps. We assessed the sensitivity of AdaTrans to varying the length of historical time steps, denoted as h . The results of this evaluation are presented in Figure 4. We observed an initial improvement in the performance of AdaTrans as h increased. However, when h exceeded a value of 3, there was a subsequent decline in performance. A possible explanation for this trend is the strong correlation of transfer time with recent traffic conditions. Introducing longer historical time steps can inadvertently add noise in the form of uncorrelated features, which then negatively impacts the model's performance. Furthermore, using extended historical data also increases the computational burden for each training epoch; specifically, the processing time rose sharply from 23 seconds to 155 seconds. Taking into account both computational efficiency and model performance, we concluded that an optimal setting for h is 3.

4.5.2 Embedding dimensions. We proceeded to analyze the impact of the embedding dimension on the performance of AdaTrans. The

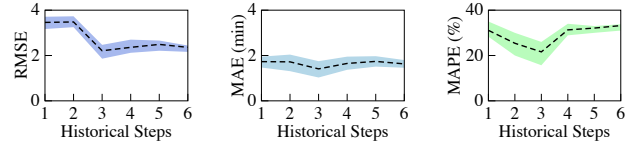


Figure 4: Performance on different historical steps. The performance increases at the beginning and then decreases when h is set larger than 3.

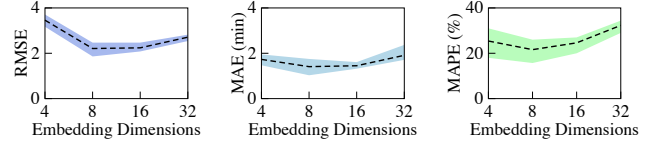


Figure 5: Performance on different embedding dimensions. The performance increases at the beginning and then decreases when it is set larger than 8.

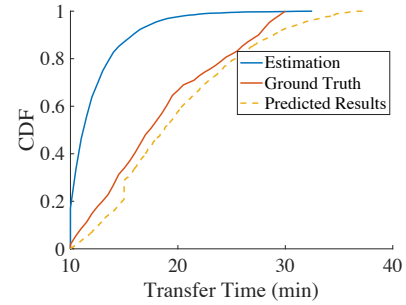


Figure 6: Visualization for transfer time prediction. The results predicted by AdaTrans are closer to the ground truth.

findings, as depicted in Figure 5, indicate that our model reaches its peak performance with an embedding dimension of 8. Notably, both too-small and excessively-large embedding dimensions are detrimental to the performance of AdaTrans. While a larger embedding dimension can capture more information, it simultaneously increases the total number of parameters. This escalation complicates the model's optimization process and can predispose the model to overfit, a phenomenon corroborated by previous research [3].

4.6 Results Visualization

To gain deeper insights into the performance of AdaTrans, we visualized the distributions of its predictions alongside the ground truth and straightforward estimation [24, 25, 27], which is observed in Figure 6. The blue solid line showcases the distribution of transfer time estimates derived from straightforward estimation [24, 25, 27] (elaborated in Sec. 4), whereas the red solid line delineates the distribution of the ground truth. Notably, the yellow dashed line highlights the distribution of AdaTrans's predictions, which align more closely with the ground truth than the blue solid line. This serves as a testament to the effectiveness of AdaTrans.

5 Related Work

This section introduces related studies and provides a detailed analysis, highlighting how our work differs from existing research.

5.1 Uni-modal Travel Time Prediction

There are two categories of existing solutions for the estimation of travel time for uni-modal transportation [6, 26]. The first one is the route-free methods [13, 20, 23, 35], which only considers the origin and destination. MURAT [23] designed an origin-destination travel time estimation and produced meaningful representation, and leveraged the underlying road network and the spatiotemporal prior knowledge. CTTE [13] fused GPS traces, smartphone inertial data, and road networks within a deep recurrent neural network. DeepI2T [20] learned the travel time mainly from the built environment images, i.e., the morphological layout image.

The second category is the route-based methods [9, 10, 12, 17, 25, 32, 33, 36, 43], which split the trip into several road segments and predict the time in each segment. DeepTTE [32] designed an end-to-end deep learning framework to estimate the travel time of the whole path directly. WDR [33] formulated ETA as a spatial-temporal regression problem and jointly trained wide linear models, deep neural networks, and recurrent neural networks. However, these designs are complex and require high computation costs. To solve the limitation, BusTr [4] and CompactETA [12] designed a simple and efficient model, respectively. Meanwhile, ConSTGAT [10] introduced a synchronous spatial-temporal module for synchronous spatial and temporal relationships, and HetETA [17] translated road maps into a multi-relational heterogeneous network.

However, most existing frameworks [13, 20, 23] that are designed for uni-modal transportation are not applicable to our scenario because they primarily model how the real-time traffic conditions affect the travel speed and then travel time. In contrast, we focus on multi-modal transportation scenarios that are affected by a wider range of factors, including both internal and external factors. For instance, we need to account for the specific characteristics of downstream transportation modes, such as the stable interval of arrival times for subways compared to buses, even during rush hours. Additionally, we must consider the impact of demand and supply on transfer time, as passengers might experience longer transfer times if the transport is full. Finally, unlike uni-modal transportation, our scenario involves multiple downstream stations for passengers to choose from, with choices potentially varying between different periods like rush and non-rush hours.

5.2 Multi-modal Route Recommendation

Currently, research in the realm of multi-modal transportation primarily centers on route recommendation [25, 27, 37–39]. This line of inquiry seeks to furnish travelers with optimal route suggestions that encompass various modes of transportation. Hydra [27] integrates uni-modal and multi-modal routes and heterogeneous urban data. HMTRL [25] studies the unified route representation learning and the semantic coherence of the historical routes.

Different from route recommendation, the problem studied in this paper is more focused on predicting the transfer time, whereas, in route recommendation papers, the transfer time is pre-calculated with assumptions, e.g., the passengers follow the recommended

route shown on maps and the average walking speed during transferring. Moreover, the methods above highly depend on the predefined or static self-learned graph for multi-modal connectivity representation and overlook the time-specific latent connectivity hidden between some stations.

5.3 Adaptive Learning

Motivated by adaptive learning [3, 7, 22], which avoids the predefined graph. AGCN [22] designed a task-driven adaptive graph that is learned for each graph data while training. This method is based on distance metrics. AGCRN [3] leveraged a node-specific pattern learning for adaptive graph representation. AGE [7] introduced a novel attributed graph embedding framework for attributed graph embedding.

However, even though the above methods break the limits of static adjacency matrix, i.e., predefined or self-learned, to learn spatial correlation, the relationships between nodes are still not time-variant. In order to solve this limitation, we design a spatial-temporal adaptive learning component and an attribute learning component that adapts to explore the uncertain external and inherent factors, respectively. Compared with previous models that adopt a self-learned static adjacent matrix, it collectively learns high-dimension relationships, i.e., the time-specific spatial correlation between two transportation modes.

6 Conclusions

In this study, we present AdaTrans, an innovative adaptive graph learning model tailored for personalized transfer time prediction within multi-modal transportation. Distinctly, AdaTrans seamlessly integrates both external and internal factors, emphasizing the correlations between environmental influences and individual preferences. To model external factors like real-time traffic, demand, and supply dynamics, we have crafted a spatial-temporal adaptive learning component. Concurrently, we deploy an attribute learning module to discern patterns in internal factors, focusing primarily on passengers' preferences. At last, we have embedded an aggregation component, leveraging a capsule network, ensuring a dynamic fusion of the influences of the aforementioned factors on transfer time. We tested AdaTrans using two real-world datasets obtained from Shenzhen City for August 2018 and September 2019. The empirical evidence strongly suggests that AdaTrans outperforms the state-of-the-art baselines. Moving forward, we aspire to adapt AdaTrans to cities of varying magnitudes, seeking to evaluate its proficiency in forecasting transfer times within diverse multi-modal transportation frameworks.

Acknowledgments

We thank all the reviewers for their insightful feedback to improve this paper. This work is partially supported by the National Science Foundation under Grant No. 2411151, 2047822, 1952096, and 1951890.

References

- [1] 2021. Bus routes. [EB/OL]. https://en.wikipedia.org/wiki/List_of_bus_routes_in_Shenzhen#Peak-time_Route_series.
- [2] 2022. Line Graph. [EB/OL]. https://en.wikipedia.org/wiki/Line_graph.
- [3] L Bai, L Yao, C Li, X Wang, and C Wang. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *NeurIPS*.
- [4] Richard Barnes, Senaka Buthpitiya, James Cook, Alex Fabrikant, Andrew Tomkins, and Fangzhou Xu. 2020. BusTr: Predicting Bus Travel Times from Real-Time Traffic. In *KDD*. 3243–3251.
- [5] Chu Cao, Zhidan Liu, Mo Li, Wenqiang Wang, and Zheng Qin. 2018. Walkway discovery from large scale crowdsensing. In *IPSN*. 13–24.
- [6] Zebin Chen, Xiaolin Xiao, Yue-Jiao Gong, Jun Fang, Nan Ma, Hua Chai, and Zhiguang Cao. 2022. Interpreting Trajectories from Multiple Views: A Hierarchical Self-Attention Network for Estimating the Time of Arrival. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) (*KDD '22*). Association for Computing Machinery, New York, NY, USA, 2771–2779. <https://doi.org/10.1145/3534678.3539051>
- [7] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. 2020. Adaptive graph encoder for attributed graph embedding. In *KDD*. 976–985.
- [8] Graham Currie. 2005. The demand performance of bus rapid transit. *Journal of public transportation* 8, 1 (2005), 3.
- [9] Longchao Da, Rohan Chhibba, Rushabh Jaiswal, Ariane Middel, and Hua Wei. 2024. Shaded Route Planning Using Active Segmentation and Identification of Satellite Images. *arXiv preprint arXiv:2407.13689* (2024).
- [10] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. 2020. Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps. In *KDD*. 2697–2705.
- [11] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics New York.
- [12] Kun Fu, Fanlin Meng, Jieping Ye, and Zheng Wang. 2020. Compacteta: A fast inference system for travel time prediction. In *KDD*. 3337–3345.
- [13] Ruipeng Gao, Xiaoyu Guo, Fuyong Sun, Lin Dai, Jiayan Zhu, Chenxi Hu, and Haibo Li. 2019. Aggressive driving saves more time? Multi-task learning for customized travel time estimation.. In *IJCAI*. 1689–1696.
- [14] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *AAAI*, Vol. 33. 3656–3663.
- [15] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*, Vol. 33. 922–929.
- [16] Marcus Handte, Muhammad Umer Iqbal, Stephan Wagner, Wolfgang Apolinarski, Pedro José Marrón, Eva María Muñoz Navarro, Santiago Martínez, Sara Izquierdo Barthelemy, and Mario González Fernández. 2014. Crowd Density Estimation for Public Transport Vehicles. In *EDBT/ICDT Workshops*. 315–322.
- [17] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. 2020. Heteta: heterogeneous information network embedding for estimating time of arrival. In *KDD*. 2444–2454.
- [18] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [19] Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention Is Not Only a Weight: Analyzing Transformers with Vector Norms. In *EMNLP*. 7057–7075.
- [20] Wuwei Lan, Yanyan Xu, and Bin Zhao. 2019. Travel time estimation without road networks: an urban morphological layout representation approach. *arXiv preprint arXiv:1907.03381* (2019).
- [21] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM*. 2615–2623.
- [22] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. 2018. Adaptive graph convolutional neural networks. In *AAAI*, Vol. 32.
- [23] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *KDD*. 1695–1704.
- [24] Hao Liu, Jindong Han, Yanjie Fu, Yanyan Li, Kai Chen, and Hui Xiong. 2023. Unified route representation learning for multi-modal transportation recommendation with spatiotemporal pre-training. *The VLDB Journal* 32, 2 (2023), 325–342.
- [25] Hao Liu, Jindong Han, Yanjie Fu, Jingbo Zhou, Xinjiang Lu, and Hui Xiong. 2020. Multi-modal transportation recommendation with unified route representation learning. *PVLDB* 14, 3 (2020), 342–350.
- [26] Hao Liu, Wenzhao Jiang, Shui Liu, and Xi Chen. 2023. Uncertainty-Aware Probabilistic Travel Time Prediction for On-Demand Ride-Hailing at DiDi. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) (*KDD '23*). Association for Computing Machinery, New York, NY, USA, 4516–4526. <https://doi.org/10.1145/3580305.3599925>
- [27] Hao Liu, Yongxin Tong, Panpan Zhang, Xinjiang Lu, Jianguo Duan, and Hui Xiong. 2019. Hydra: A personalized and context-aware multi-modal transportation recommendation system. In *KDD*. 2314–2324.
- [28] Wenjun Lyu, Guang Wang, Yu Yang, and Desheng Zhang. 2021. Mover: Generalizability Verification of Human Mobility Models via Heterogeneous Use Cases. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–21.
- [29] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. 2010. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS one* 5, 4 (2010), e10047.
- [30] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in neural information processing systems*. 3856–3866.
- [31] Heng Tan, Yukun Yuan, Shuxin Zhong, and Yu Yang. 2023. Joint Rebalancing and Charging for Shared Electric Micromobility Vehicles with Human-system Interaction. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*. 235–236.
- [32] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. 2018. When will you arrive? estimating travel time based on deep neural networks. In *AAAI*.
- [33] Zheng Wang, Kun Fu, and Jieping Ye. 2018. Learning to estimate the travel time. In *KDD*. 858–866.
- [34] Zhaoyang Wang, Beihong Jin, Fusang Zhang, Ruiyang Yang, and Qiang Ji. 2017. Exploiting trip patterns in passenger trajectory streams for bus scheduling optimization in real time. In *MDM*. IEEE, 266–271.
- [35] Hua Wei, Chacha Chen, Chang Liu, Guanjie Zheng, and Zhenhui Li. 2021. Learning to simulate on sparse trajectory data. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part IV*. Springer, 530–545.
- [36] Hua Wei, Dongkuan Xu, Junjie Liang, and Zhenhui Li. 2021. How do we move: Modeling human movement with system dynamics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4445–4452.
- [37] Cai Xu, Ziyu Guan, Wei Zhao, Hongchang Wu, Yunfei Niu, and Beilei Ling. 2019. Adversarial incomplete multi-view clustering.. In *IJCAI*, Vol. 7. 3933–3939.
- [38] Cai Xu, Jiajun Si, Ziyu Guan, Wei Zhao, Yue Wu, and Xiyue Gao. 2024. Reliable conflictive multi-view learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16129–16137.
- [39] Cai Xu, Wei Zhao, Jinglong Zhao, Ziyu Guan, Xiangyu Song, and Jianxin Li. 2022. Uncertainty-aware multiview deep learning for internet of things applications. *IEEE Transactions on Industrial Informatics* 19, 2 (2022), 1456–1466.
- [40] Guang Yang, Yuequn Zhang, Jinquan Hang, Xinyue Feng, Zejun Xie, Desheng Zhang, and Yu Yang. 2023. CARPG: Cross-City Knowledge Transfer for Traffic Accident Prediction via Attentive Region-Level Parameter Generation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2939–2948.
- [41] Jingfeng Yang, Jian Gang Jin, Jianjun Wu, and Xi Jiang. 2017. Optimizing passenger flow control and bus-bridging service for commuting metro lines. *Computer-Aided Civil and Infrastructure Engineering* 32, 6 (2017), 458–473.
- [42] Bing Yu, Haoqiang Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *IJCAI*. 3634–3640.
- [43] Guanjie Zheng, Chang Liu, Hua Wei, Porter Jenkins, Chacha Chen, Tao Wen, and Zhenhui Li. 2021. Knowledge-based Residual Learning.. In *IJCAI*.
- [44] Shuxin Zhong, Wenjun Lyu, Desheng Zhang, and Yu Yang. 2022. Bikecap: Deep spatial-temporal capsule network for multi-step bike demand prediction. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 831–841.
- [45] Shuxin Zhong, William Yubeaton, Wenjun Lyu, Guang Wang, Desheng Zhang, and Yu Yang. 2023. RLIFE: Remaining Lifespan Prediction for E-scooters. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 3544–3553.