Article:
Huang, H., Mao, J., Liu, R. orcid.org/0000-0003-0627-3184 et al. (3 more authors) (2024) MTLMetro: A Deep Multi-Task Learning Model for Metro Passenger Demands Prediction. IEEE Transactions on Intelligent Transportation Systems. ISSN 1524-9050

https://doi.org/10.1109/tits.2024.3373565

MTLMetro: A Deep Multi-task Learning Model for Metro Passenger Demands Prediction

Hao Huang, Jiannan Mao, Ronghui Liu, Weike Lu, Tianli Tang, Lan Liu

***Abstract*—Accurate prediction of passenger demand is essential for the efficient operation and management of metro systems. In practical scenarios, strategies to enhance metro service quality often require passenger demand information on multiple fronts, such as inflow to a station, outflow from a station, as well as transition flow between entry/exit stations. While predictions for a single type of passenger demand have been extensively studied, limited attention was paid to jointly predicting multiple demands. This problem is challenging due to the complex relationships among multiple demands (e.g., inflow is only correlated with historical inflow, while the outflow is not only correlated with outflow but also determined by the inflow) and the imbalanced training issue of multiple prediction tasks. To address these challenges, this paper proposes a deep multi-task learning (MTL) model called MTLMetro to co-predict multiple demands in metro systems. More specifically, we deploy the message-passing schemes in graph neural networks (GNNs) as the knowledge-sharing mechanisms in the MTL model to capture the inherent relationships among multiple demands. To balance the training of multiple tasks, we introduce a novel weighting scheme named dynamic weight average (DWA), which can dynamically adapt relative weight for each task. In addition, the partial observability problem of transition flow is also considered in MTLMetro in an end-to-end manner. Empirical evaluation on a real-world dataset demonstrates MTLMetro's superior performance across the different demand prediction tasks when compared to several benchmarks. Further ablation experiments verify the effectiveness of the proposed modules and the weighting method.**

***Index Terms*—Metro demand prediction, multi-task learning, deep learning, graph neural network, dynamic weight average.**

This work was supported in part by the National Natural Science Foundation of China (No.61873216, 62103292, 71890972, and 71890970), and in part by the Science and Technology Project of Sichuan Province (No.2020YFSY0020). The work of Hao Huang was supported by China Scholarship Council (No. 202207000074). The work of Tianli Tang was supported by the project of Jiangsu Funding Program for Excellent Postdoctoral Talent (No. 2022ZB114), the project of 'Chunhui Jihua' of Ministry of Education, China (No. HZKY20220156), and the National Natural Science Foundation of China (No. 52311530090/42261144745). *(Corresponding author: Lan Liu, Ronghui Liu.)*

Hao Huang, Jiannan mao, and Lan Liu are with the School of Transportation and Logistics, and National and Local Joint Engineering Laboratory of Integrated Intelligent Transportation, Southwest Jiaotong University, Chengdu, 610031, China. (e-mail: haohuang_h@163.com; jiannan_mao@hotmail.com; jianan_l@home.swjtu.edu.cn).

Ronghui Liu is with the Institute for Transport Studies, University of Leeds, Leeds LS2 9JT, United Kingdom. (e-mail: R.Liu@its.leeds.ac.uk).

Weike Lu is with the School of Rail Transportation, Soochow University, Suzhou, 21513, China; Alabama Transportation Institute, 248 Kirkbride Lane, Tuscaloosa, AL 35487, USA. (e-mail: wklu@suda.edu.cn).

Tianli Tang is with the School of Transportation, Southeast University, Nanjing, 211189, China. (e-mail: T-Tang@seu.edu.cn).

I. Introduction

RAIL-based metro system, emblematic of high-capacity, environmentally-efficient transport, represents a fundamental cornerstone in mitigating traffic congestion and traffic pollution, key challenges to sustainable urbanization. However, many metropolises now suffer substantial pressure from excessive passenger demand on their metro systems. For instance, over 10 million passengers use the Beijing metro system daily, surpassing the system capacity, particularly during peak hours. This necessitates strategies such as train scheduling [1] and passenger inflow control [2] to alleviate the oversaturated situation. To support such measures, accurate and short-term prediction of passenger flows through the system is essential, which could benefit the effective operation and management of metro systems.

Metro systems primarily encompass two types of passenger demand: i) node demand, denoting the number of passengers entering/exiting a station during a time interval (termed as inflow/outflow), and ii) transition demand, representing the number of passengers traversing between entrance-exit stations during a time interval (termed as origin-destination (OD) flow). These two measures of passenger demand are interrelated but play different roles in the operation and management of metro systems. For example, the node demand is integral to passenger inflow control at stations, whereas the transition demand is used in planning and scheduling the train services.

Recently, extensive studies have been proposed for short-term demand prediction in metro systems, using methods that range from classical statistical methods to advanced artificial intelligence (AI) methods (e.g., machine learning and deep learning) [3], [4], [5], [6], [7], [8]. Among the AI methods, deep learning models, such as convolutional neural networks (CNNs) [9], [10], [11], [12], recurrent neural networks (RNNs) [13], [14], [15], [16], and graph neural networks (GNNs) [17], [18], [19], are favored by researchers, owing to their ability to model complex dependencies embedded in metro systems. However, most of these studies focus on predicting a single type of passenger demand, either node or transition demand.

Notice that, for some operation strategies, multiple demands are simultaneously considered. For example, both inflow and OD flow are considered when collaboratively optimizing the train timetable and passenger flow control strategies [20]. Thus, a palpable need emerges for accurate co-prediction of multiple demands in metro systems - an area has received limited research attention thus far. Due to the complex

intercorrelation across demands, traditional machine learning and deep learning models cannot effectively address the joint learning of multiple tasks. Fortunately, multi-task learning (MTL) is a promising machine learning paradigm that leverages inherent information across multiple related tasks to improve performance [21]. Recent studies on the co-prediction of multiple demands in metro systems have verified the superiority of this paradigm [22], [23], [24], [25]. However, there are two challenges yet to be comprehensively addressed in existing MTL models:

Challenge 1: *How to share knowledge across tasks (knowledge-sharing mechanism).* By leveraging useful knowledge among related tasks, an MTL model has the potential to improve performance for all tasks [26]. Clearly, the multiple demands in metro systems are closely related, e.g., the flow conservation relationship (a station's outflow equals the sum of its OD flow). From the model architecture point-of-view, designing delicate knowledge-sharing mechanisms derived from the inherent relationships among multiple demands is a key challenge for an MTL model.

Challenge 2: *How to tackle the joint learning of multiple tasks (task balancing).* Taking the weighted sum of the sub-task loss as the loss function is a typical way to train an MTL model. However, as the convergence speed and training difficulty for different tasks may differ, one or more tasks can dominate the overall model training, potentially resulting in bias against less-weighted tasks [26]. For a better overall performance, more attention should be paid to the challenging tasks, while less training focus should be allocated to the easier ones. Manual tuning of loss weights is tedious, so adaptively tuning the loss weights is highly desirable to balance the joint training of multiple tasks.

Beyond the co-prediction problems, another two challenges are also concerned with demand prediction in metro systems:

Challenge 3: *Leveraging message-passing schemes in GNNs.* GNNs, endowed with the message-passing scheme, enable the exchange of information/messages on node and edge levels [27]. Examples of the message-passing scheme in GNNs for node- and edge-featured graphs are shown in **Fig.1**, from which we can observe that information can be passed directly from nodes to nodes or edges. The exchange of information can be seen as a form of knowledge-sharing process. Thus, we argue that the message-passing schemes in GNNs can be further designed as knowledge-sharing mechanisms in an MTL model.

Challenge 4: *Partial observability of transition flow.* Another practical challenge for passenger demand prediction in metro systems is the partial observability problem of transition flow [28], i.e., the automatic fare collection (AFC) system cannot record completed transition data until passengers finish their journeys. In other words, the true transition demand at the current time can only be obtained after a time lag. When predicting multiple passenger demands in metro systems, neglecting the partial observability issue and using the transition demand observed in the future as the inputs of prediction models is unrealistic in practice, while directly using the incomplete transition demand may cause a loss of massive information. Thus, integrating the transition demand completion task into the demand prediction model is also worth studying.

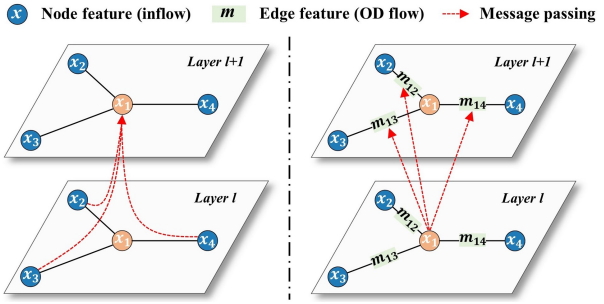


Fig.1 Message-passing scheme in GNNs for node-featured (left) and edge-featured (right) graphs

With the above concerns, we propose a novel deep MTL model called MTLMetro to simultaneously predict multiple demands in metro networks. Firstly, we propose an OD completion (ODC) module to address the partial observability problem of transition demand and provide richer information for subsequent prediction tasks. Secondly, three tailored GNNs are carefully designed to characterize the inherent relationships among multiple demands in MTLMetro. Finally, a dynamic adaptive weighting method named dynamic weight average (DWA) is introduced to balance the joint learning of all tasks. Compared with the existing studies, the main contributions of this paper are summarized as follows:

(1) We propose a novel deep learning model, the MTLMetro, to co-predict the inflow, outflow, and transition flow in metro systems. Completing the transition demand is involved as an auxiliary task in MTLMetro to address the partial observability issue. A dynamic loss weighting method is introduced for MTLMetro to adapt the relative importance of each task during training, allowing better overall performance for all tasks.

(2) We design three GNNs, in which the tailored message-passing schemes, derived from the inherent relationships among multiple demands, are developed as the knowledge-sharing mechanisms in MTLMetro. In addition to capturing spatial dependencies underlying metro systems, the proposed GNNs can be employed to leverage beneficial information across associated tasks.

(3) We demonstrate that the proposed MTLMetro outperforms other benchmark models by comparative experiments on a real-world metro demand dataset in Chengdu, China. Further experiments shed light on the superiority of the components in MTLMetro.

The rest of the paper is arranged as follows: Section II reviews the related literature. We provide some preliminaries of the study in Section III. Section IV introduces the formulation process and methodologies of the proposed model in detail. The dataset, benchmarks, experiment results, and model interpretation are presented in Section V. Finally, Section VI concludes with a summary of the findings of the case study.

II. Related works

In this section, we conduct an extensive literature review on short-term passenger demand prediction under two categories: single-type demand and multiple demands prediction.

A. Single-type demand prediction

Over the past few decades, numerous single-type metro passenger demand (e.g., inflow, outflow, and OD flow) prediction models have been developed. Early studies focused on predicting the node demand using methods such as auto-regression integrated moving average (ARIMA) [3] and generalized autoregressive conditional heteroskedasticity (GARCH) [4]. Since the passenger demand data shows a strong non-linearity, traditional statistical models may fail to capture the complex features. Machine learning models have proved to be more efficient in metro passenger demand prediction. Some successful machine learning studies include support vector machine (SVM) [5], dynamic Bayesian networks [6], radial basis function networks [7], and neural networks (NNs) [8]. However, most studies could not well capture the inherent patterns of passenger demands.

Recent studies have shown that deep learning models can achieve superior prediction performance than traditional models and machine learning models because of their ability to capture spatial-temporal correlations [29], [30], [31], [32]. In particular, RNNs, especially their variants, e.g., long-short term network (LSTM) [14], [15] and gated recurrent unit (GRU) [16], were employed to detect the temporal dependencies of metro inflows/outflows. CNNs were used to learn the spatial dependencies of passenger demand [9], [10], [11], [12]. However, the prerequisite to applying CNNs is transitioning passenger demand data to regular Euclidean structures, which may violate the non-Euclidean nature of metro networks. Owning the ability to model non-Euclidean structures, GNNs have been leveraged to seize the non-Euclidean spatial dependencies and solve the drawbacks of CNNs [17]. Combining GNNs with RNNs has been proven to be an efficient way to capture the spatial-temporal dependencies embedded in inflows/outflows [18], [19], [33].

In terms of metro OD flow prediction, there are only a few studies on this topic. Noursalehi *et al.* [9] combined CNN with convolutional LSTM (Conv-LSTM) to forecast the future metro OD flows. Zhang *et al.* [34] designed an operation-oriented deep-learning model called the spatiotemporal convolutional neural network (STCNN) to realize short-term OD flow prediction. Some researchers have attempted to address this task with GNNs to better model the spatial dependencies. For example, Jiang *et al.* [28] introduced a temporally shifted graph convolution (TSGC) to model the lagged temporal relationships among OD pairs. Liu *et al.* [35] modeled the metro network as graphs based on realistic topology, passenger flow similarity, and correlation. A graph convolution gated recurrent unit (GC-GRU) was then proposed to incorporate these graphs. However, the above GNN-based models are limited to node-level operations and fail to incorporate the edge features containing important information. When converting the metro networks into graphs, taking the OD flows as the edge features is intuitive and valid [36]. Therefore, we aim to develop GNNs for edge-featured graphs to further exploit spatial-temporal dependencies of transition demand.

The partial observability issue of OD flow information has attracted increasing attention in recent years. To address this problem, some studies suggested using additional information, such as inflow, outflow, historical OD flow, and unfinished order information, to replace the unavailable OD matrices as the inputs of prediction models [10], [37], [38]. Others attempted to estimate the completed OD. For example, Ye *et al.* [39] utilized inflow and historical OD distributions and proposed an NN-based model to complete the OD matrices. Jiang *et al.* [28] developed a reconstruction mechanism in a deep leaning-based model, taking inflow and partially observed OD flow as inputs to estimate the OD flow. Using the real-time demand information (e.g., inflow, outflow, destination allocation of inflow, origin allocations of outflow), Zheng *et al.* [40] established a multi-view passenger flow (MVPF) model to learn the latent representation of OD flow. Although multiple demand sources are utilized to deal with the partial observability issue, the previous studies are limited to predicting the OD flow, neglecting the inherent correlations between these multiple demands. Recently, Xu et al. [24] and Liu *et al.* [25] integrated the OD completion task with multiple demand prediction tasks using the MTL paradigm, and the results demonstrated the superiority of the MTL paradigm in enhancing overall performance. Detailed comparisons between the above two MTL models and our proposed model will be discussed in the following subsection.

B. Multiple demands prediction

MTL is a machine learning paradigm that can leverage useful information in multiple tasks to help learn a more accurate model for each task. As a promising AI technical, MTL has been used in several domains, such as recommendation systems [41], natural language processing [42], and computer vision [43].

In the transportation domain, substantial research efforts are dedicated to jointly predicting multiple states via MTL. For an MTL model, one key is designing the model architecture where the information can be transmitted across tasks, i.e., the knowledge-sharing mechanism. A common way is directly concatenating [22], [44], summing [44], or sharing [23], [24] the representations of different tasks. Some modified operators were proposed to obtain deeper information across tasks. For instance, Liu *et al.* [25] introduced an MTL model for metro transition demands prediction, where the Transformer was used to propagate the mutual information among demands. Zhang *et al.* [45] proposed an MTL model with LSTM to catch the correlations between multiple demands. Zhang *et al.* [46] constructed an MTL model for ride-hailing demand prediction, in which the task correlation was estimated by dynamic time warping (DTW). Feng *et al.* [47] developed an MTL model for co-prediction of ride-hailing demands, in

which a matrix-factorized module was employed to decode representations for each task separately. Ke *et al.* [48] established an MTL model that imposes a prior tensor normal distribution on the weights of different task networks. Liang *et al.* [49] were the first to leverage the message-passing schemes in GNNs to capture the cross-mode spatiotemporal dependencies in multimodal transportation systems. However, the message-passing scheme employed in [49] is unsuitable for predicting multiple demands in metro systems for two primary reasons. Firstly, they employed GNNs that cannot utilize spatial dependencies on the edge level, thus limiting their ability to address transition-based prediction tasks. Secondly, the inherent relationships among passenger demands in metro systems are more complex than those captured by geographic and semantic correlations in [49].

Moreover, designing the loss function is as important as designing the architecture for an MTL model. A general and handy way is to set the loss function as the sum [22], [45], [46], [48], weighted sum [24], [25], [44], [47], [49], or automatic weighted sum [23] of sub-task loss. Various tasks may converge at different speeds, resulting in an imbalance of training for sub-tasks. For better overall performance, the loss function in the MTL model should be designed to enable the tasks to be trained at the same pace. However, existing studies in the transportation demand prediction domain have not addressed the task-balancing problem in MTL models.

TABLE I compares the studies related to MTL in the transportation prediction domain regarding the above-listed challenges, including the knowledge-sharing mechanism (challenge 1), task balancing (challenge 2), message-passing level (challenge 3), and partial observability of transition flow (challenge 4). Four major highlights can be summarized: i) Few studies fully considered the potential relationships among demands when designing the knowledge-sharing mechanism in MTL models; ii) Most studies neglected the task-balancing issue in MTL models; iii) Few studies extended the message-passing scheme in GNNs on the edge level; iv) the partial observability of transition flow problem of metro transition demand has not been thoroughly studied in the MTL paradigm. To address these issues, this paper aims to: i) Explore the connections between message-passing schemes in GNNs and knowledge-sharing mechanisms in MTL models, leveraging inherent relationships among passenger demands; ii) Develop a loss weighting method that can adapt the task weights over time to ensure balanced task training; iii) Generalize message-passing schemes in GNNs on both node and edge levels, exploiting spatial-temporal dependencies of node and transition demands; iv) Integrate the transition demand completion as an auxiliary task within the proposed MTL model to provide richer information for the prediction tasks.

III. PRELIMINARIES

A. Basic definitions

Definition 1: *Inflow and outflow vectors*. The total time period is first portioned into T time slots by a given time interval. At timestamp $t, t \in \{1,\cdots,T\}$, the inflow $x_{in,n}^t$ and outflow $x_{out,n}^t$ represent the cumulative number of passengers entering and exiting the station $n, n \in \{1,\cdots,N\}$, where N denotes the number of stations. The inflow and outflow can be represented by vectors $\boldsymbol{X}_{in}^t = \left[x_{in,1}^t,\cdots,x_{in,N}^t\right] \in \mathbb{R}^N$ and $\boldsymbol{X}_{out}^t = \left[x_{out,1}^t,\cdots,x_{out,N}^t\right] \in \mathbb{R}^N$.

Definition 2: *OD matrixes*. Here, we define two kinds of transition demands, as in Eq. (1). At timestamp t, the inbound-based OD (INOD) matrix is denoted by $\boldsymbol{M}_{in}^t \in \mathbb{R}^{N\times N}$, where each entry $m_{in,i,j}^t$ denotes the transition demand entering station i heading to station j. The outbound-based OD (OUTOD) matrix is denoted by $\boldsymbol{M}_{out}^t \in \mathbb{R}^{N\times N}$, and each entry $m_{out,i,j}^t$ represents the transition demand reaching station j at time t originating from station i.

$$\boldsymbol{M}_{in}^t = \begin{bmatrix} m_{in,1,1}^t & m_{in,1,2}^t & \cdots & m_{in,1,N}^t \\ m_{in,2,1}^t & m_{in,2,2}^t & \cdots & m_{in,2,N}^t \\ \vdots & \vdots & \vdots & \vdots \\ m_{in,N,1}^t & m_{in,N,2}^t & \cdots & m_{in,N,N}^t \end{bmatrix}$$
$$\boldsymbol{M}_{out}^t = \begin{bmatrix} m_{out,1,1}^t & m_{out,1,2}^t & \cdots & m_{out,1,N}^t \\ m_{out,2,1}^t & m_{out,2,2}^t & \cdots & m_{out,2,N}^t \\ \vdots & \vdots & \vdots & \vdots \\ m_{out,N,1}^t & m_{out,N,2}^t & \cdots & m_{out,N,N}^t \end{bmatrix} \quad (1)$$

TABLE I
COMPARISON OF RELATED STUDIES

Publication	Goal	Knowledge-sharing mechanism	Task Balancing	Message-passing level	Partial observability
[44]	Taxi demands	Sum/Concatenation			
[45]	Taxi demands	LSTM			
[46]	Ride-hailing demands	DTW			
[47]	Ride-hailing demands	Matrix factorized			
[48]	Ride-hailing demands	Prior tensor distribution			
[22]	Metro demands	Concatenation			
[23]	Metro demands	Parameter sharing			
[24]	Metro demands	Parameter sharing			✓
[25]	Metro demands	Transformer			✓
[49]	Multimodal demands	Geographic/semantic-based message-passing scheme		Node level	
This paper	Metro demands	Inherent relationships-based message-passing scheme	✓	Node and edge levels	✓

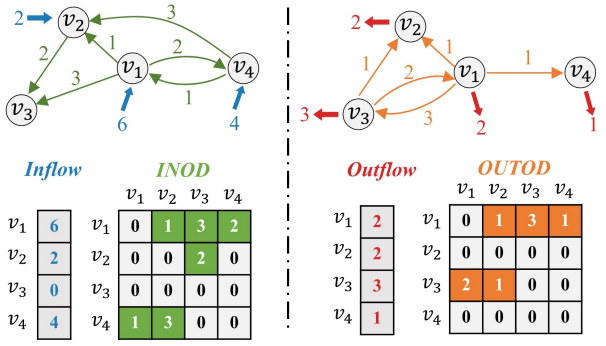


Fig.2 Flows in a simple metro network

Fig.2 presents flows in a simple metro network. Intuitively, flow conservation relationships exist between multiple demands. For example, the sum of the INOD originating from station *i* equals the inflow of the station:

$$x_{in,i}^{t} = \sum_{j=1}^{N} m_{in,i,j}^{t} \tag{2}$$

Similarly, the sum of the OUTOD flow reaching station *j* equals the outflow of the station:

$$x_{out,j}^{t} = \sum_{i=1}^{N} m_{out,i,j}^{t} \tag{3}$$

One should note that the INOD matrix cannot be observed in real-time since a travel time gap exists between passengers entering the origin station and exiting the destination station. In other words, the INOD matrix can only be completed and obtained until the passengers have finished their trips [50]. On the contrary, the OUTOD matrix can be fully observed in real-time since the observed passengers have reached their destination. Based on the above analysis and inspired by [39], we then define real-time/delayed inflow/INOD.

Definition 3: *Real-time/delayed inflow vectors*. Given two timestamps τ and $t,(\tau \le t)$, the real-time inflow is denoted by $\boldsymbol{X}_{in}^{\tau,t,r}$, representing the demand entering a station at τ that can reach destinations before or at t. Then, we define the delayed inflow $\boldsymbol{X}_{in}^{\tau,t,d}$ as the demand that will finish their trips after t. Obviously, $\boldsymbol{X}_{in}^{\tau,t,r} + \boldsymbol{X}_{in}^{\tau,t,d} = \boldsymbol{X}_{in}^{\tau}$.

Definition 4: *Real-time/delayed INOD matrixes*. Given two timestamps τ and $t,(\tau \le t)$, the real-time INOD matrix is denoted by $\boldsymbol{M}_{in}^{\tau,t,r} \in \mathbb{R}^{N\times N}$, representing the transition demand originating from τ that can reach the destinations before or at t. On the other hand, the remainders will finish their trips after t, and this kind of transition demand is defined as the delayed INOD matrix denoted by $\boldsymbol{M}_{in}^{\tau,t,d} \in \mathbb{R}^{N\times N}$. Obviously, $\boldsymbol{M}_{in}^{\tau,t,r} + \boldsymbol{M}_{in}^{\tau,t,d} = \boldsymbol{M}_{in}^{\tau}$.

Two graphs are then proposed to model the metro networks based on the defined passenger demands. A topology graph that uses the topological structure of the metro networks is introduced to describe the non-Euclidean spatial relationships. Besides, unlike the topology graph without edge features, we propose an edge-featured OD graph to leverage the information of transition demands.

Definition 5: *Topology Graph.* A topology graph $\boldsymbol{G}_{tg} = (\boldsymbol{V}, \boldsymbol{E}_{tg}, \boldsymbol{A}_{tg})$ is proposed to model the physical structure of the metro network, where $\boldsymbol{V}$ is a set of nodes representing the stations in the metro network, and $\boldsymbol{E}_{tg}$ denotes the edges. $\boldsymbol{A}_{tg} \in \mathbb{R}^{N\times N}$ is the adjacency matrix describing the topological connectedness of nodes, in which the entry $[\boldsymbol{A}_{tg}]_{i,j} = 1$ if nodes $v_i \in \boldsymbol{V}$ and $v_j \in \boldsymbol{V}$ are physically connected; otherwise $[\boldsymbol{A}_{tg}]_{i,j} = 0$. The neighborhood of v_i is defined as $Ne_{tg}(i) = \{j \,|\, [\boldsymbol{A}_{tg}]_{i,j} = 1\}$. In the topology graph, each node is associated with the inflow/outflow features, while edges are with no weights.

Definition 6: *OD graph.* An OD graph $\boldsymbol{G}_{od} = (\boldsymbol{V}, \boldsymbol{E}_{od}, \boldsymbol{A}_{od})$ is defined to involve the transition demand. $\boldsymbol{V}$ is the node set denoting the stations and $\boldsymbol{E}_{od}$ is the edge set. Different from $\boldsymbol{G}_{tg}$ that simply considers the physical connections, $\boldsymbol{G}_{od}$ takes the station's reachability into account. The reachability $r_{i,j}$ denotes whether passengers originating from station $v_i \in \boldsymbol{V}$ can reach station $v_j \in \boldsymbol{V}$ via the metro network, i.e., $r_{i,j} = 1$ if v_j is reachable from v_i; otherwise $r_{i,j} = 0$. Then, each entry of the adjacency matrix $\boldsymbol{A}_{od}$ is defined according to the reachability $[\boldsymbol{A}_{od}]_{i,j} = r_{i,j}$. The neighborhood of node v_i in the OD graph is then defined as $Ne_{od}(i) = \{j \,|\, [\boldsymbol{A}_{od}]_{i,j} = 1\}$. In the OD graph, each node is featured by the inflow/outflow, and each edge is featured by the INOD/OUTOD.

B. Research problem

In general, a passenger demand prediction task is a time-series prediction problem that uses historical observations to predict future passenger demand. In this paper, four kinds of passenger demand, i.e., inflow, outflow, INOD, and OUTOD, are involved. To fully leverage inherent information among these demands, we formulate an MTL model to jointly predict passenger demands. Notably, considering the partial observability problem of INOD, data completion is viewed as an auxiliary task in the proposed MTL model. Thus, the research problem of this paper is formally defined as follows.

Multiple passenger demands prediction. Given *K*-step historical passenger demands, $[\boldsymbol{X}_{in}^{t-K+1}, \cdots, \boldsymbol{X}_{in}^{t}]$, $[\boldsymbol{X}_{out}^{t-K+1}, \cdots, \boldsymbol{X}_{out}^{t}]$, $[\boldsymbol{X}_{in}^{t-K+1,t,r}, \cdots, \boldsymbol{X}_{in}^{t,t,r}]$, $[\boldsymbol{X}_{in}^{t-K+1,t,d}, \cdots, \boldsymbol{X}_{in}^{t,t,d}]$, $[\boldsymbol{M}_{in}^{t-K+1,t,r}, \cdots, \boldsymbol{M}_{in}^{t,t,r}]$, $[\boldsymbol{M}_{out}^{t-K+1}, \cdots, \boldsymbol{M}_{out}^{t}]$, we develop an MTL model to complete $[\boldsymbol{M}_{in}^{t-K+1}, \cdots, \boldsymbol{M}_{in}^{t}]$, and collectively predict the inflow $\boldsymbol{X}_{in}^{t+1}$, outflow $\boldsymbol{X}_{out}^{t+1}$, INOD $\boldsymbol{M}_{in}^{t+1}$, and OUTOD $\boldsymbol{M}_{out}^{t+1}$ in the future.

IV. Methodology

Fig.3 briefly presents the framework of the proposed MTLMetro for co-prediction of multiple demands in metro systems. In our proposed approach, we first develop an ODC

module to address the partial observability problem of INOD. Then, three edge-featured GNNs are developed to capture the inherent spatial correlations among demands. By carefully designing message-passing schemes in the GNNs, useful knowledge can be leveraged and shared across tasks. Besides, the proposed model also considers a GRU module to capture the temporal dependencies. Finally, a dynamic loss weighting method, DWA, is introduced to train MTLMetro and address the task balancing problem.

A. ODC module for INOD completion

In practice, the travel time gap leads to the partial observability problem of INOD. In this subsection, we aim to develop an ODC module to complete INOD.

According to definition 4, the INOD matrix $\boldsymbol{M}_{in}^{\tau}$ can be decomposed into a real-time INOD matrix $\boldsymbol{M}_{in}^{\tau,t,r}$ and a delayed INOD matrix $\boldsymbol{M}_{in}^{\tau,t,d}$. Since $\boldsymbol{M}_{in}^{\tau,t,r}$ can be observed in real-time, the estimation of $\boldsymbol{M}_{in}^{\tau}$ mainly depends on the estimation of $\boldsymbol{M}_{in}^{\tau,t,d}$. Note that $\boldsymbol{M}_{in}^{\tau,t,d}$ can also be represented by the product of $\boldsymbol{X}_{in}^{\tau,t,d}$ and the corresponding delayed passenger distribution probability $\boldsymbol{P}_{in}^{\tau,d} \in \mathbb{R}^{N\times N}$, that is, $\boldsymbol{M}_{in}^{\tau,t,d} = \boldsymbol{X}_{in}^{\tau,t,d} * \boldsymbol{P}_{in}^{\tau,d}$, where $*$ denotes the broadcast operation. Because $\boldsymbol{X}_{in}^{\tau,t,d}$ can be easily obtained according to definition 3, the problem of estimating $\boldsymbol{M}_{in}^{\tau,t,d}$ can be come down to estimate $\boldsymbol{P}_{in}^{\tau,d}$. In reality, passenger flow in metro systems shows an apparent weekly period [28], [34], [39]. We assume that the delayed distribution probability at the same time of the previous week $\bar{\boldsymbol{P}}_{in}^{\tau,d}$ can be employed as the estimation of $\boldsymbol{P}_{in}^{\tau,d}$. In practice, $\bar{\boldsymbol{P}}_{in}^{\tau,d}$ can be easily obtained according to historical observations.

Based on the above analysis, a simple feedforward neural network (FNN)-based ODC module is proposed, taking the real-time INOD, delayed inflow, and probability of delayed INOD as inputs. The details of ODC are shown in **Fig.4 (a)**. The process of ODC is as follows:

$$\hat{\boldsymbol{m}}_{in,i,j}^{\tau} = \boldsymbol{m}_{in,i,j}^{\tau,t,r} + \sigma(\boldsymbol{x}_{in,i}^{\tau,t,d}\bar{p}_{in,i,j}^{\tau,d}\boldsymbol{W}_{in} + \boldsymbol{b}_{in}) \tag{4}$$

where $\boldsymbol{m}_{in,i,j}^{\tau,t,r} \in \mathbb{R}^{C}$ represents the real-time INOD from v_i to v_j, $\boldsymbol{x}_{in,i}^{\tau,t,d} \in \mathbb{R}^{C}$ denotes the delayed inflow of v_i, and C is the feature dimension. $\bar{p}_{in,i,j}^{\tau,d}$ is the delayed distribution probability of OD from v_i to v_j at the time τ of previous week. $\boldsymbol{W}_{in} \in \mathbb{R}^{C\times C}$ is the trainable matrix, and $\boldsymbol{b}_{in} \in \mathbb{R}^{C}$ represents the trainable bias, $\sigma(\cdot)$ is the non-linear activation function. $\hat{\boldsymbol{m}}_{in,i,j}^{\tau} \in \mathbb{R}^{C}$ is the completion result of INOD from v_i to v_j at time τ. Thus, the completed INOD for all OD pairs can be represented as $\hat{\boldsymbol{M}}_{in}^{\tau} = \{\hat{\boldsymbol{m}}_{in,i,j}^{\tau}\}_{i,i=1}^{N}, \hat{\boldsymbol{M}}_{in}^{\tau} \in \mathbb{R}^{N\times N\times C}$.

Different from [39], which separated the INOD completion task from the subsequent prediction tasks, we treat it as an auxiliary task in MTLMetro so that it can be trained in an end-to-end manner.

B. Edge-featured GNNs for knowledge-sharing across passenger demands

The core idea of GNNs lies in the message-passing scheme that enables exchanging and aggregating information among nodes/edges. In the context of MTL, it aims to boost overall learning performance by leveraging and sharing beneficial knowledge across related tasks. When comparing GNNs to the MTL paradigm, the message-passing in GNNs can be seen as

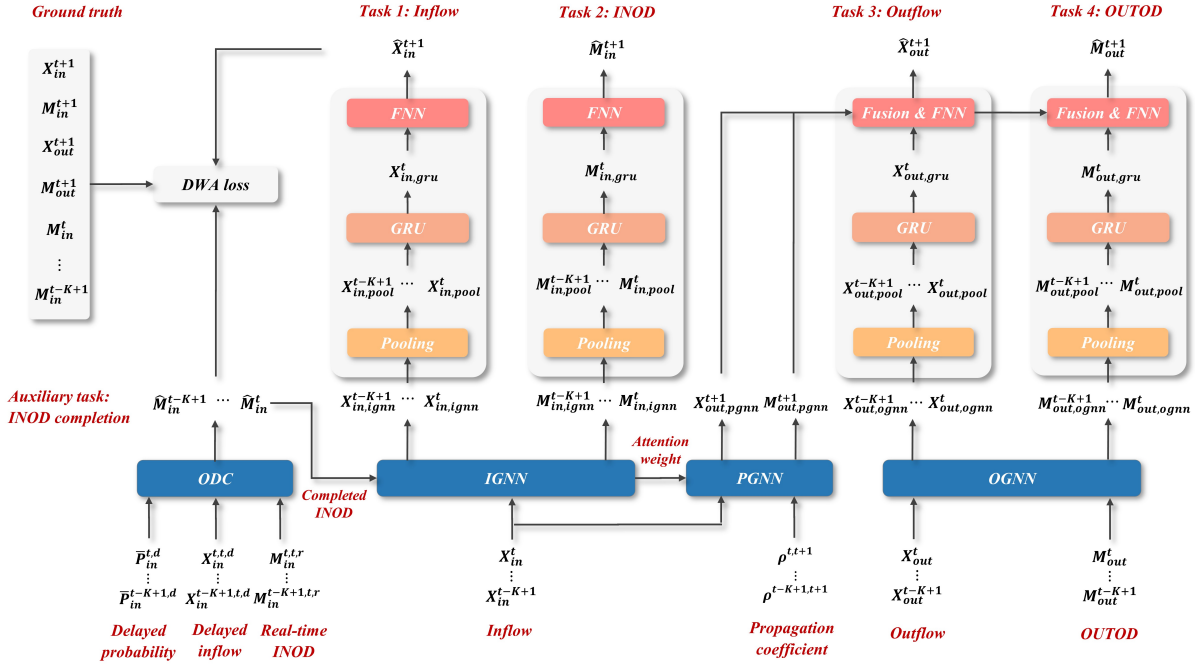


Fig.3 Framework of the proposed MTLMetro

a form of knowledge-sharing in MTL.

Designing the knowledge-sharing mechanism of an MTL model necessitates meticulous attention to the inter-task relationships. To capture the inherent correlations among multiple demands, this subsection introduces three GNNs with customized message-passing schemes tailored to these relationships. The following parts will introduce the details of the three modules.

1) IGNN

Recognizing that the INOD represents the spatial distribution of the inflow, we propose an inbound graph neural network (IGNN) module with an in-to-od message-passing scheme to capture this spatial relationship. The details of IGNN are shown in **Fig.4 (b)**. For a historical time $\tau, \tau \in \{t-K+1, \cdots, t\}$, given the inflow features $\boldsymbol{X}_{in}^{\tau} = \{\boldsymbol{x}_{in,i}^{\tau}\}_{i=1}^{N}$ and completed INOD features $\widehat{\boldsymbol{M}}_{in}^{\tau} = \{\widehat{\boldsymbol{m}}_{in,i,j}^{\tau}\}_{i,j=1}^{N}$, the process of the IGNN module is as follows:

$$\boldsymbol{x}_{in,i}^{\prime\tau} = \sigma(\sum\nolimits_{j} \boldsymbol{x}_{in,j}^{\tau} \boldsymbol{W}_{in,g} + \boldsymbol{b}_{in,g}), j \in Ne_{tg}(i) \tag{5}$$

$$z_{i,j}^{\tau} = \sigma([(\boldsymbol{x}_{in,i}^{\tau} \boldsymbol{W}_{in,v} + \boldsymbol{b}_{in,v}) \,\|\, (\widehat{\boldsymbol{m}}_{in,i,j}^{\tau} \boldsymbol{W}_{in,e} + \boldsymbol{b}_{in,e})] \boldsymbol{W}_{a}), j \in Ne_{od}(i) \tag{6}$$

$$\alpha_{i,j}^{\tau} = \frac{\exp(z_{i,j}^{\tau})}{\sum_{j} \exp(z_{i,j}^{\tau})}, j \in Ne_{od}(i) \tag{7}$$

$$\overline{\boldsymbol{m}}_{in,i,j}^{\tau} = \alpha_{i,j}^{\tau} \sigma(\boldsymbol{x}_{in,i}^{\tau} \boldsymbol{W}_{in,v} + \boldsymbol{b}_{in,v}), j \in Ne_{od}(i) \tag{8}$$

$$\boldsymbol{m}_{in,i,j}^{\prime\tau} = \beta_{in} \overline{\boldsymbol{m}}_{in,i,j}^{\tau} + (1-\beta_{in}) \sigma(\widehat{\boldsymbol{m}}_{in,i,j}^{\tau} \boldsymbol{W}_{in,e} + \boldsymbol{b}_{in,e}), j \in Ne_{od}(i) \tag{9}$$

where $\boldsymbol{x}_{in,i}^{\tau} \in \mathbb{R}^{C}$ and $\boldsymbol{x}_{in,j}^{\tau} \in \mathbb{R}^{C}$ denote the inflow features of nodes v_i and v_j at time τ, respectively. $\widehat{\boldsymbol{m}}_{in,i,j}^{\tau} \in \mathbb{R}^{C}$ represents the completed INOD features of edge $e_{i,j}$ at τ. $\boldsymbol{W}_{in,q} \in \mathbb{R}^{C \times O}, q \in \{g,v,e\}$ and $\boldsymbol{b}_{in,q} \in \mathbb{R}^{O}, q \in \{g,v,e\}$ are the trainable weights and bias, and O represents the feature dimension. $[\cdot \,\|\, \cdot]$ is the concatenation operation. $\boldsymbol{W}_{a} \in \mathbb{R}^{2O \times 1}$ denotes the linear transformation that can map a matrix to a scaler. $z_{i,j}^{\tau}$ and $\alpha_{i,j}^{\tau}$ represent the unnormalized and normalized attention score of v_i to $e_{i,j}$, and $\beta_{in} \in [0,1]$ is a learnable weight. $\overline{\boldsymbol{m}}_{in,i,j}^{\tau}$ is the latent node feature. $\boldsymbol{x}_{in,i}^{\prime\tau} \in \mathbb{R}^{O}$ and $\boldsymbol{m}_{in,i,j}^{\prime\tau} \in \mathbb{R}^{O}$ represent the output features for v_i and $e_{i,j}$ of the IGNN module, respectively. For all nodes and edges, the output features are defined as $\boldsymbol{X}_{in,ignn}^{\tau} = \{\boldsymbol{x}_{in,i}^{\prime\tau}\}_{i=1}^{N}, \boldsymbol{X}_{in,ignn}^{\tau} \in \mathbb{R}^{N \times O}$ and $\boldsymbol{M}_{in,ignn}^{\tau} = \{\boldsymbol{m}_{in,i,j}^{\prime\tau}\}_{i,j=1}^{N}, \boldsymbol{M}_{in,ignn}^{\tau} \in \mathbb{R}^{N \times N \times O}$, respectively.

In the proposed IGNN, Eq. (5) is used to output the node features, while Eq. (6-9) is employed to produce the edge features. Specifically, a node-to-node message-passing scheme [51] is developed to operate on the topology graph to capture the spatial dependencies between inflows, as Eq. (5) shows. Then, the in-to-od message-passing scheme based on the graph attention networks (GAT) [52] is designed to operate on the OD graph to model the relationship between inflow and

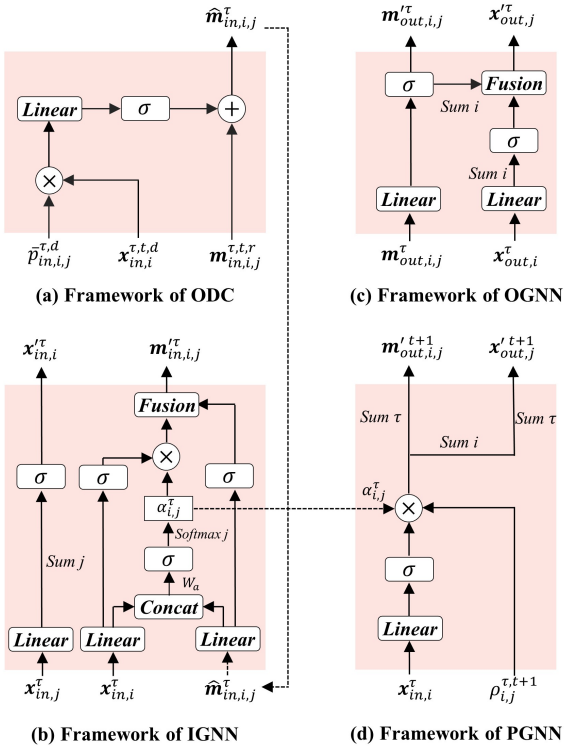


Fig.4 Framework of the proposed modules

INOD. In Eq. (6), linear transformations $\boldsymbol{W}_{in,v}$ and $\boldsymbol{W}_{in,e}$ are initially employed to project inflow and INOD representations into high-dimensional spaces. A concatenation operator is used to fuse the inflow and INOD representations. Subsequently, a trainable linear transformation $\boldsymbol{W}_{a}$ is applied to compute the pair-wise attention score $z_{i,j}^{\tau}$, which signifies the influence from the inflow of v_i to the INOD of $e_{i,j}$. In Eq. (7), a softmax operation is applied to normalize the attention score and make coefficients comparable across different INOD pairs. Once the normalized attention score is obtained, a simple product of the attention score and inflow is conducted to determine the influence of inflow on INOD, as shown in Eq. (8). Notice that we can get $\sum_{j} \alpha_{i,j}^{\tau} = 1, j \in Ne_{od}(i)$ according to Eq. (7). The flow conservation relationships between inflow and INOD will be guaranteed by Eq. (8). In Eq. (9), a fusion operator with a learnable weight β_{in} is developed to adaptively integrate the task-across node knowledge and task-specific edge knowledge when updating the edge features.

In contrast to the traditional GAT, which is primarily designed for graphs with node features, the attention mechanism in IGNN is specifically devised for graphs that incorporate both node and edge features.

2) OGNN

The outflow at a station emerges as the aggregate outcome of the corresponding OUTOD, and we develop an outbound graph neural network (OGNN) module equipped with an od-to-out message-passing scheme to model their relationship. The details of OGNN are shown in **Fig.4 (c)**. For a historical

time $\tau, \tau \in \{t-K+1, \cdots, t\}$, given the outflow features $\boldsymbol{X}_{out}^{\tau} = \{\boldsymbol{x}_{out,i}^{\tau}\}_{i=1}^{N}$ and OUTOD features $\boldsymbol{M}_{out}^{\tau} = \{\boldsymbol{m}_{out,i,j}^{\tau}\}_{i,j=1}^{N}$, the process of the OGNN module is as follows:

$$\bar{\boldsymbol{x}}_{out,j}^{\tau} = \sigma(\sum\nolimits_i \boldsymbol{x}_{out,i}^{\tau} \boldsymbol{W}_{out,g} + \boldsymbol{b}_{out,g}), i \in Ne_{tg}(j) \tag{10}$$

$$\boldsymbol{m}'^{\tau}_{out,i,j} = \sigma(\boldsymbol{m}_{out,i,j}^{\tau} \boldsymbol{W}_{out,e} + \boldsymbol{b}_{out,e}) \tag{11}$$

$$\breve{\boldsymbol{x}}_{out,j}^{\tau} = \sum\nolimits_i \boldsymbol{m}'^{\tau}_{out,i,j}, i \in Ne_{od}(j) \tag{12}$$

$$\boldsymbol{x}'^{\tau}_{out,j} = \beta_{out} \bar{\boldsymbol{x}}_{out,j}^{\tau} + (1-\beta_{out}) \breve{\boldsymbol{x}}_{out,j}^{\tau} \tag{13}$$

where $\boldsymbol{x}_{out,i}^{\tau} \in \mathbb{R}^C$ and $\boldsymbol{x}_{out,j}^{\tau} \in \mathbb{R}^C$ are the outflow features of v_i and v_j at τ, respectively. $\boldsymbol{m}_{out,i,j}^{\tau} \in \mathbb{R}^C$ represents the OUTOD features of $e_{i,j}$ at τ. $\boldsymbol{W}_{out,q} \in \mathbb{R}^{C \times O}, q \in \{g,e\}$ and $\boldsymbol{b}_{out,q} \in \mathbb{R}^O, q \in \{g,e\}$ are the trainable weights and bias, $\beta_{out} \in [0,1]$ is the learnable coefficient. $\bar{\boldsymbol{x}}_{out,j}^{\tau}$ and $\breve{\boldsymbol{x}}_{out,j}^{\tau}$ are the latent node features. $\boldsymbol{x}'^{\tau}_{out,j} \in \mathbb{R}^O$ and $\boldsymbol{m}'^{\tau}_{out,i,j} \in \mathbb{R}^O$ represent the output features for node v_j and edge $e_{i,j}$, respectively.

In Eq. (10), the spatial dependencies of outflows are detected by the node-to-node message-passing scheme [51]. Specifically, similar to the classic GCN, when generating the features of node v_j, the information from the neighborhoods $v_i, i \in Ne_{tg}(j)$ of the topology graph is utilized. A FNN is then employed to map the edge features to high-dimensional space, as Eq. (11) shows. Eq. (12) shows a concise sum-based od-to-out message-passing scheme operated on the OD graph to model the flow conservation relationship between outflow and OUTOD. After that, the information derived from the OUTOD, i.e., $\breve{\boldsymbol{x}}_{out,j}^{\tau}$, together with the updated node features $\bar{\boldsymbol{x}}_{out,j}^{\tau}$, is fused to output the node features $\boldsymbol{x}'^{\tau}_{out,j}$ in Eq. (13). The output features for all nodes and edges are defined as $\boldsymbol{X}_{out,ognn}^{\tau} = \{\boldsymbol{x}'^{\tau}_{out,i}\}_{i=1}^{N}, \boldsymbol{X}_{out,ognn}^{\tau} \in \mathbb{R}^{N \times O}$ and $\boldsymbol{M}_{out,ognn}^{\tau} = \{\boldsymbol{m}'^{\tau}_{out,i,j}\}_{i,j=1}^{N}, \boldsymbol{M}_{out,ognn}^{\tau} \in \mathbb{R}^{N \times N \times O}$, respectively.

3) PGNN

Considering that the outflow and OUTOD can be interpreted as the propagation results of the inflow, we propose a propagation graph neural network (PGNN) module featuring an in-to-out message-passing scheme to capture the propagation relationship.

When modeling the relationship between inflow and outflow/OUTOD, the first step is figuring out which previous inflow of other stations will influence the target station's outflow/OUTOD at the target time. Here, we carefully propose a time-varying propagation coefficient to describe the propagation relationship. At time t, given a historical time $\tau, \tau \in \{t-K+1, \cdots, t\}$ and a future time $\boldsymbol{\tau}$, the propagation coefficient $\rho_{i,j}^{\tau,\boldsymbol{\tau}}$ is defined as:

$$\rho_{i,j}^{\tau,\boldsymbol{\tau}} = \begin{cases} 1, (\boldsymbol{\tau} - \tau - 1) \times \Delta \le \mu_{i,j} < (\boldsymbol{\tau} - \tau + 1) \times \Delta \,|\, i \ne j \\ 0, \text{otherwise} \end{cases} \tag{14}$$

where $\mu_{i,j}$ denotes the average travel time between v_i and v_j, Δ is the length of the time interval. Intuitively, $\rho_{i,j}^{\tau,\boldsymbol{\tau}} = 1$ if the passenger origin from v_i at time τ can arrive at v_j at time $\boldsymbol{\tau}$, else $\rho_{i,j}^{\tau,\boldsymbol{\tau}} = 0$.

Based on the propagation coefficient, we then proposed the PGNN module. The details of PGNN are shown in **Fig.4 (d)**. At time t, we set $\tau \in \{t-K+1, \cdots, t\}$ and $\boldsymbol{\tau} = t+1$, given the inflow features $\boldsymbol{X}_{in}^{\tau} = \{\boldsymbol{x}_{in,i}^{\tau}\}_{i=1}^{N}$ and the propagation coefficient $\boldsymbol{\rho}^{\tau,t+1} = \{\rho_{i,j}^{\tau,t+1}\}_{i,j=1}^{N}$, the process of the PGNN module is as follows:

$$\boldsymbol{m}_{out,i,j}^{\tau,t+1} = \sigma(\boldsymbol{x}_{in,i}^{\tau} \boldsymbol{W}_{in,p}^{\tau} + \boldsymbol{b}_{in,p}^{\tau}) \rho_{i,j}^{\tau,t+1} \alpha_{i,j}, j \in Ne_{od}(i) \tag{15}$$

$$\boldsymbol{x}_{out,j}^{\tau,t+1} = \sum\nolimits_i \boldsymbol{m}_{out,i,j}^{\tau,t+1}, i \in Ne_{od}(j) \tag{16}$$

$$\boldsymbol{m}'^{t+1}_{out,i,j} = \sum_{\tau=t-K+1}^{t} \boldsymbol{m}_{out,i,j}^{\tau,t+1} \tag{17}$$

$$\boldsymbol{x}'^{t+1}_{out,j} = \sum_{\tau=t-K+1}^{t} \boldsymbol{x}_{out,j}^{\tau,t+1} \tag{18}$$

where $\boldsymbol{W}_{out,p}^{\tau} \in \mathbb{R}^{C \times O}$ and $\boldsymbol{b}_{out,p}^{\tau} \in \mathbb{R}^O$ denote the trainable weights and bias. $\alpha_{i,j}^{\tau}$ represents the learned attention score in the IGNN module. $\boldsymbol{x}'^{t+1}_{out,j} \in \mathbb{R}^O$ and $\boldsymbol{m}'^{t+1}_{out,i,j} \in \mathbb{R}^O$ represent the output outflow features for node v_j and INOD features for edge $e_{i,j}$, respectively. $\boldsymbol{X}_{out,pgnn}^{t+1} = \{\boldsymbol{x}'^{t+1}_{out,i}\}_{i=1}^{N}, \boldsymbol{X}_{out,pgnn}^{t+1} \in \mathbb{R}^{N \times O}$ and $\boldsymbol{M}_{out,pgnn}^{t+1} = \{\boldsymbol{m}'^{t+1}_{out,i,j}\}_{i,j=1}^{N}, \boldsymbol{M}_{out,pgnn}^{t+1} \in \mathbb{R}^{N \times N \times O}$ denote the features for all nodes and edges, respectively.

The PGNN module introduces an in-to-out message-passing scheme based on the OD graph. In Eq. (15), the inflow features are first transformed into high-level dimensions and then multiplied by the propagation coefficient $\rho_{i,j}^{\tau,t+1}$ to judge whether the inflow will influence the corresponding outflow/OUTOD. The attention score $\alpha_{i,j}^{\tau}$ learned by the IGNN module is then used to determine how much influence should be considered. The flow conservation relationships between outflow and OUTOD will be guaranteed by Eq. (16). A sum operator is applied to obtain the total influence of the historical inflow on the OUTOD and outflow, as defined in Eq. (17) and Eq. (18)

C. GRU for temporal dependencies modeling

RNNs are widely used deep learning models for short-term passenger demand prediction because of their ability to capture temporal dependencies. Among RNNs, LSTM and GRU are the most popular RNN variants. Here, we employ GRU since it is as powerful as LSTM but with fewer parameters. Given a K-step input sequence $\{\boldsymbol{x}_{\tau}\}_{\tau=t-K+1}^{t}$, details of the GRU are as follows [53]:

$$z_\tau = \sigma(\boldsymbol{W}_z \boldsymbol{x}_\tau + \boldsymbol{U}_z \boldsymbol{h}_{\tau-1} + \boldsymbol{b}_z) \tag{19}$$

$$r_\tau = \sigma(\boldsymbol{W}_r \boldsymbol{x}_\tau + \boldsymbol{U}_r \boldsymbol{h}_{\tau-1} + \boldsymbol{b}_r) \tag{20}$$

$$\bar{\boldsymbol{h}}_\tau = \tanh(\boldsymbol{W}_h \boldsymbol{x}_\tau + \boldsymbol{U}_h (r_\tau * \boldsymbol{h}_{\tau-1}) + \boldsymbol{b}_h) \tag{21}$$

$$\boldsymbol{h}_\tau = z_\tau \boldsymbol{h}_{\tau-1} + (1 - z_\tau)\bar{\boldsymbol{h}}_\tau \tag{22}$$

where $\boldsymbol{x}_\tau$ denotes the input at time τ. z_τ and r_τ are the update gate and reset gate, respectively. The reset gate controls the extent to which information from the previous time step is disregarded, whereas the update gate determines the extent to which information from the previous time step is retained. $\boldsymbol{W}_z$, $\boldsymbol{W}_r$, $\boldsymbol{W}_h$, $\boldsymbol{U}_z$, $\boldsymbol{U}_r$ and $\boldsymbol{U}_h$ are weight matrices, $\boldsymbol{b}_z$, $\boldsymbol{b}_r$ and $\boldsymbol{b}_h$ are biases. $\boldsymbol{h}_\tau$ presents the output at time τ, which is obtained based on the current input $\boldsymbol{x}_\tau$ and previous output $\boldsymbol{h}_{\tau-1}$. In the proposed model, the input $\boldsymbol{x}_\tau$ of the GRU module could be the inflow, outflow, and OD flow.

D. MTLMetro for multiple demands prediction

The MTLMetro is proposed to co-predict multiple demands in metro networks, including inflow, outflow, INOD, and OUTOD. In MTLMetro, predicting each demand is viewed as a sub-task, and completing INOD is regarded as an auxiliary task. The framework of MTLMetro is shown in **Fig. 3**.

At time *t,* given the *K*-step historical inputs, the completed INOD can be obtained by the ODC module as $\widehat{\mathsf{M}}_{in}^t = [\widehat{\boldsymbol{M}}_{in}^{t-K+1}, \cdots, \widehat{\boldsymbol{M}}_{in}^t] \in \mathbb{R}^{K \times N \times N \times C}$. The IGNN module outputs the inflow features $\mathsf{X}_{in,ignn}^t = [\boldsymbol{X}_{in,ignn}^{t-K+1}, \cdots, \boldsymbol{X}_{in,ignn}^t] \in \mathbb{R}^{K \times N \times O}$ and INOD features $\mathsf{M}_{in,ignn}^t = [\boldsymbol{M}_{in,ignn}^{t-K+1}, \cdots, \boldsymbol{M}_{in,ignn}^t] \in \mathbb{R}^{K \times N \times N \times O}$. The OGNN module generates the outflow and OUTOD features as $\mathsf{X}_{out,ognn}^t = [\boldsymbol{X}_{out,ognn}^{t-K+1}, \cdots, \boldsymbol{X}_{out,ognn}^t] \in \mathbb{R}^{K \times N \times O}$ and $\mathsf{M}_{out,ognn}^t = [\boldsymbol{M}_{out,ognn}^{t-K+1}, \cdots, \boldsymbol{M}_{out,ognn}^t] \in \mathbb{R}^{K \times N \times N \times O}$, respectively. Besides, the PGNN module will output the outflow features $\boldsymbol{X}_{out,pgnn}^{t+1} \in \mathbb{R}^{N \times O}$ and OUTOD features $\boldsymbol{M}_{out,pgnn}^{t+1} \in \mathbb{R}^{N \times N \times O}$ of the target time $t+1$.

Before feeding the outputs of the IGNN and OGNN into the GRU layers, a max-pooling operation is applied to reduce the dimension of the node features and edge features. Taking $\boldsymbol{X}_{in,ignn}^t$ as an example, the pooling process is set as:

$$\boldsymbol{X}_{in,pool}^t = maxpool(\boldsymbol{X}_{in,ignn}^t) \tag{23}$$

where $\boldsymbol{X}_{in,pool}^t \in \mathbb{R}^N$ denotes the output of the pooling layer.

The pooled sequence $\mathsf{X}_{in,pool}^t = [\boldsymbol{X}_{in,pool}^{t-K+1}, \cdots, \boldsymbol{X}_{in,pool}^t] \in \mathbb{R}^{K \times N}$ is then fed into the GRU layer:

$$\boldsymbol{X}_{in,gru}^t = \mathrm{GRU}(\mathsf{X}_{in,pool}^t) \tag{24}$$

where $\boldsymbol{X}_{in,gru}^t \in \mathbb{R}^O$ is the output of the last time step in the GRU layer.

Similarly, we can obtain corresponding outputs of the GRU layer for outflow, INOD, and OUTOD as $\boldsymbol{X}_{out,gru}^t \in \mathbb{R}^O$, $\boldsymbol{M}_{in,gru}^t \in \mathbb{R}^{N \times O}$, and $\boldsymbol{M}_{out,gru}^t \in \mathbb{R}^{N \times O}$, respectively.

A fusion operator is then used to fuse the outputs of the GRU layer and PGNN layer:

$$\boldsymbol{X}_{out,fusion}^t = \beta_{fusion,1} \boldsymbol{X}_{out,gru}^t + (1 - \beta_{fusion,1}) \boldsymbol{X}_{out,pgnn}^{t+1} \tag{25}$$

$$\boldsymbol{M}_{out,fusion}^t = \beta_{fusion,2} \boldsymbol{M}_{out,gru}^t + (1 - \beta_{fusion,2}) \boldsymbol{M}_{out,pgnn}^{t+1} \tag{26}$$

where $\beta_{fusion,1}$ and $\beta_{fusion,2}$ are the learnable coefficients.

Finally, $\boldsymbol{X}_{in,gru}^t$, $\boldsymbol{M}_{in,gru}^t$, $\boldsymbol{X}_{out,fusion}^t$, and $\boldsymbol{M}_{out,fusion}^t$ are fed into FNN layers to obtain the final prediction results $\widehat{\boldsymbol{X}}_{in}^{t+1}$, $\widehat{\boldsymbol{X}}_{out}^{t+1}$, $\widehat{\boldsymbol{M}}_{in}^{t+1}$, and $\widehat{\boldsymbol{M}}_{out}^{t+1}$.

E. DWA for task balancing

In the previous subsections, we present the details of the MTLMetro model. In addition to network architecture, the training method is also important for an MTL model. In this paper, a weighted sum of sub-task loss is used to train the model:

$$\begin{aligned} L(\Theta) = {} & \lambda_1 \left\| \mathsf{M}_{in} - \widehat{\mathsf{M}}_{in} \right\|_2^2 + \lambda_2 \left\| \boldsymbol{X}_{in}^{t+1} - \widehat{\boldsymbol{X}}_{in}^{t+1} \right\|_2^2 + \lambda_3 \left\| \boldsymbol{M}_{in}^{t+1} - \widehat{\boldsymbol{M}}_{in}^{t+1} \right\|_2^2 \\ & + \lambda_4 \left\| \boldsymbol{X}_{out}^{t+1} - \widehat{\boldsymbol{X}}_{out}^{t+1} \right\|_2^2 + \lambda_5 \left\| \boldsymbol{M}_{out}^{t+1} - \widehat{\boldsymbol{M}}_{out}^{t+1} \right\|_2^2 \end{aligned} \tag{27}$$

where Θ are all learnable parameters in the proposed model, and $\lambda_i, i \in \{1, \cdots, 5\}$ are weight coefficients of the sub-tasks.

For an MTL model, the training difficulty of tasks may differ, leading to a scenario where one or more tasks dominate the model training. Thus, efforts should be made to avoid the imbalance in training to improve overall performance. We then develop a task weighting method, DWA [54], that can adapt the loss weights over time to balance the joint learning of multiple tasks. Based on the DWA, the weights λ_i in Eq. (27) are then replaced by time-varying ones as:

$$\lambda_i(\eta) = \frac{5 \exp(w_i(\eta - 1)/P)}{\sum_i \exp(w_i(\eta - 1)/P)} \tag{28}$$

$$w_i(\eta - 1) = \frac{l_i(\eta - 1)}{l_i(\eta - 2)} \tag{29}$$

where $l_i(\eta - 1)$ denotes the loss value of the task i at iteration $\eta - 1$, and $w_i(\eta - 1)$ represents the relative descending speed of task i at iteration $\eta - 1$. The softmax operator of Eq. (28) multiplying 5 is applied to obtain $\lambda_i(\eta)$ and ensure $\sum_i \lambda_i(\eta) = 5$. P represents the temperature that controls the softness of weighting in the softmax operator. For $\tau = 1, 2$, we set $w_i(\eta) = 1$.

V. CASE STUDY

A. Data and benchmarks

The real-world dataset collected from the AFC system of the Chengdu metro is used to verify the proposed model. The map (produced by © Mapbox, data by © OpenStreetMap) of the study case network in Chengdu is shown in **Fig.5**. We treat the stations in the metro network as nodes and construct graphs $\boldsymbol{G}_{tg}$ and $\boldsymbol{G}_{od}$ according to definitions 5-6. Graph-related data, such as adjacency matrices and neighborhood information, can subsequently be obtained. The dataset contains the inbound and outbound information of 6 lines and 136 stations ranging from 1st August to 31st October 2018, containing about 192 million passenger demand records. According to the real operation of Chengdu metro systems, the research time period for each day starts from 6:00 a.m. to 24:00 p.m. In this experiment, the length of the time interval is set as 10 minutes. This dataset is divided into three parts: the data from 1st August to 13th October 2018 for training, the data from 14th to 22nd October 2018 for validation, and the data from 23rd to 31st October 2018 for testing.

To demonstrate the effectiveness of the proposed model, the benchmarks considered in this paper are listed as follows:

HA: Historical Average is the most fundamental method for demand prediction. The historical average of the training dataset is used to predict passenger demands.

KNN: K-Nearest Neighbors is a classical machine learning model. The number of neighbors of KNN is tuned from 1 to 10, among which the optimal values for inflow, outflow, INOD, and OUTOD prediction are 6, 6, 7, and 8, respectively.

GBDT: Gradient Boosting Decision Tree is an ensemble machine learning model that can combine several simple tree models to achieve better performance. The maximum depth of the tree is tuned from {1, 3, 5, 7, 10, 15, 20}. The best parameters for inflow, outflow, INOD, and OUTOD prediction are 15, 10, 20, and 20, respectively.

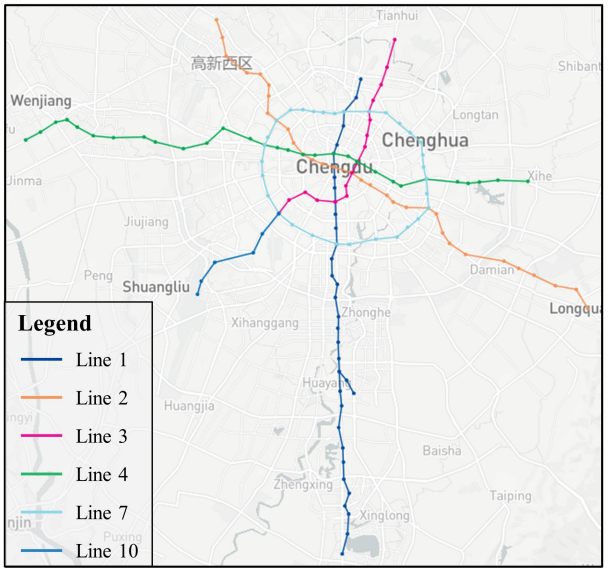


Fig.5 The map of the study case network in Chengdu, China

RF: Random Forest is an effective machine learning model for passenger demand modeling. The number of trees in the forest is selected from {10, 50, 100, 150, 200}, and it turns out that 50, 100, 150, and 150 trees make the best performance for predicting inflow, outflow, INOD, and OUTOD, respectively.

MLP: Multi-layer Perception is the basic deep learning model. We build a 4-hidden layer MLP model with ReLU as the activation function.

GRU: Due to the ability to model sequence data, GRU is employed as a baseline model. We build a 2-layer stacked GRU model in this paper.

T-GCN: Temporal Graph Convolutional Network [55] is a state-of-the-art spatial-temporal deep learning model for transportation demand prediction. A T-GCN model with a GCN layer and a GRU layer is implemented in this paper. Due to the limitation of GCN, which can only operate on node level, T-GCN is used for inflow and outflow prediction.

HIAM: Heterogeneous Information Aggregation Machine [25] is a deep learning model for short-term OD passenger flow prediction model that can simultaneously forecast the future INOD and OUTOD. HIAM serves as a baseline for jointly predicting passenger transition demand.

MR-STN: Spatio-Temporal Network framework based on Multi-Relational [56] proposes a multi-relational learning module to model the relationships among multiple demands, which can simultaneously predict passenger inflow and outflow. MR-STN can be viewed as a baseline for node demand prediction based on the MTL paradigm.

NENN: Node and Edge feature in graph Neural Networks (NENN) [57] is a recently proposed model that considers both the node and edge features in a graph, which means that it can be used for both node and transition demands prediction.

Herein, we introduce the hyper-parameter settings for all deep learning models. When training the above deep learning models, the initial learning rate is set from {0.1, 0.01, 0.001, 0.0001} with a 0.5 decay rate after every 20 training steps, the batch size is selected from {8, 16, 32, 64}, and the hidden size for each layer chosen from {16, 32}. All the deep learning models are trained for 200 epochs by an ADAM optimizer with MSE as the loss function. Besides, an early stopping strategy is applied to the validation dataset to avoid overfitting. We set K=12 , which means that the nearest 120 min of historical observations is used to predict future passenger demands. All experiments are implemented on a PC with 96G RAM and one NVIDIA 3080 GPU.

The performances of these models on the test dataset are evaluated via three commonly used metrics:

(1) Mean absolute error (MAE):

$$MAE = \frac{1}{I}\sum_{i=1}^{I}|y_i - \hat{y}_i| \tag{30}$$

(2) Root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{I}\sum_{i=1}^{I}(y_i - \hat{y}_i)^2} \tag{31}$$

(3) Mean absolute percentage error (MAPE):

$$MAPE = \frac{1}{I}\sum_{i=1}^{I}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \quad (32)$$

where y_i and $\hat{y}_i$ are the i^{th} ground truth and the prediction value, respectively, and I is the size test dataset. Given that the division by zero yields an infinite MAPE, we only consider the data records with positive values when calculating the MAPE.

B. Prediction performance comparison

The prediction performance of baselines and the proposed model are listed in TABLE II, and the best performance is marked in bold. Some interesting conclusions can be drawn as follows:

(1) The performance of HA is worse than the machine learning and deep learning models, especially for outflow prediction, because of its simple assumptions about the relationships between the data.

(2) Model tailored for time series, GRU, outperforms the classical deep learning model MLP, indicating that considering the inherent temporal dependencies may improve the prediction performance.

(3) By integrating GNNs with RNNs, T-GCN can capture both spatial and temporal dependencies embedded in node/transition demand, helping boost prediction performance.

(4) We can observe that the MTL models, HIAM and MR-STN, outperform the single-task learning models, indicating that the MTL paradigm leverages useful information in multiple tasks to enhance the accuracy of each subtask.

(5) Though NENN is capable of multi-task prediction, it does not outperform T-GCN, HIAM, and MR-STN. The reason is that NENN models the relationship between the node and edge by the traditional attention mechanism that neglects inherent correlations between node demands and transition demands. The improper modeling of the relationships between multiple demands results in the recession of NENN.

(6) For all models, it can be observed that the MAPE values for INOD/OUTOD are comparatively high and exceed those for inflow/outflow, a discrepancy primarily attributed to two factors. Firstly, MAPE, as a measure of relative error, is determined by contrasting the absolute difference between predicted and actual values against the actual ones. Therefore, the MAPE tends to produce relatively large values when the ground truths are small. Notably, INOD/OUTOD values are consistently smaller than the inflow/outflow values, as demonstrated in Eq. (2) and Eq. (3), contributing to the larger MAPE observed for the former one. Secondly, INOD/OUTOD exhibits a higher complexity than inflow/outflow, characterized by high data dimensionality and sparse spatial-temporal dependencies [10]. In the studied Chengdu metro system, which comprises 136 stations, a total of 18,496 INOD/OUTOD pairs are identified, with more than 26% exhibiting zero flow during the day. This complexity, arising from high dimensionality and spatial-temporal sparsity, poses challenges to the accurate identification of INOD/OUTOD patterns, resulting in higher MAPE errors compared to inflow/outflow. Despite the challenges, the MAPE values for the proposed model's predictions of INOD and OUTOD are 41.093% and 39.854%, respectively, both well below the 50% threshold. This threshold, as noted by Lewis [58], is recognized as an indicator of reasonable accuracy, suggesting that the proposed model is reliable for practical applications.

(7) The proposed MTLMetro model outperforms all single-task (HA, KNN, GBDT, RF, MLP, GRU, and T-GCN) and multi-task models (HIAM, MR-STN, and NENN) on all tasks, demonstrating that the proposed MTLMetro is an effective model for multiple demands prediction. The improvements of MTL come from the following aspects: i) considering the spatial and temporal dependencies by integrating GNNs and RNNs, ii) completing the INOD matrix to provide richer information, iii) an MTL architecture with carefully designed knowledge-sharing mechanisms according to the inherent relationships among tasks, and iv) an adaptive and dynamic weighting method for task balancing. Detailed model interpretations are conducted in the following subsections.

C. Experiment results visualization

In this subsection, we visualize the detailed prediction results to gain insights into the spatial-temporal patterns learned by the model.

Fig.6 presents the prediction results and the ground truth of inflows from three selected stations: Sihe Station, situated in a residential area; Chunxi Road Station, located in a central business district (CBD); Zhongba Station, positioned within a residential and industrial vicinity. Due to disparities in points of interest (POI), the inflows demonstrate diverse temporal

TABLE II
EXPERIMENT RESULTS ON MULTIPLE DEMANDS PREDICTION

	Inflow			INOD			Outflow			OUTOD		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	20.851	40.954	39.698%	0.698	1.455	69.531%	22.500	43.333	42.994%	0.707	1.499	75.384%
KNN	17.624	30.970	28.269%	0.676	1.313	68.091%	19.735	33.753	30.820%	0.596	1.145	66.670%
GBDT	17.210	33.121	26.661%	0.637	1.204	55.660%	19.689	34.242	30.762%	0.583	1.065	54.523%
RF	16.954	33.005	25.334%	0.614	1.167	54.913%	19.522	33.601	29.200%	0.584	1.070	55.027%
MLP	16.971	33.021	25.430%	0.611	1.132	54.892%	19.549	33.822	29.236%	0.595	1.100	55.588%
GRU	16.813	32.484	23.148%	0.593	1.113	52.557%	18.982	32.787	27.977%	0.565	1.072	49.146%
T-GCN	16.264	30.503	22.557%	-	-	-	18.892	31.729	27.445%	-	-	-
HIAM	-	-	-	0.535	1.002	45.442%	-	-	-	0.540	1.013	44.196%
MR-STN	16.003	29.981	22.311%	-	-	-	18.652	31.634	27.261%	-	-	-
NENN	16.792	31.128	23.138%	0.564	1.003	48.415%	19.031	32.112	28.143%	0.559	1.063	46.392%
MTLMetro	**15.617**	**29.358**	**21.975%**	**0.521**	**0.966**	**41.093%**	**18.496**	**31.289**	**26.803%**	**0.524**	**0.985**	**39.854%**

patterns, for example, morning peak, evening peak, and bimodal patterns for the inflows at Sihe, Chunxi Road, and Zhongba stations, respectively. In **Fig.6**, the black lines represent the ground truth, while the red lines depict the prediction results. From **Fig.6**, we can observe that the prediction curves trace well with the ground truth curves for all cases, indicating that the proposed model can robustly capture the temporal dependencies of the inflow well.

Fig.7 shows the ground truth and predicted values of four timestamps (7:30 a.m., 7:40 a.m., 7:50 a.m., and 8:00 a.m.) INOD of a line with 33 stations during the morning peak. In **Fig.7**, the first row represents the ground truth, while the second represents the prediction results. The color bar of **Fig.7** represents the quantity of INOD trips, with deeper shades of red indicating higher numbers of OD trips. Clearly, we can observe that the proposed model can well capture the passenger patterns embedded in INOD both temporally and spatially.

In addition to the primary demand prediction tasks, INOD completion is regarded as an auxiliary task within the proposed MTLMetro framework. **Fig.8** displays the distribution of the completion residuals, illustrating the difference between the ground truth and the completed values. The relatively small magnitude of the residuals suggests that the proposed ODC module effectively addresses the completion of partially observed INOD. Additionally, the residuals basically follow a normal distribution with zero mean, which implies the absence of significant non-random patterns in the residuals, indicating that the proposed model has adequately learned sufficient INOD information [14].

D. Effect of the proposed modules

To better understand the contribution of the modules to the performance of MTLMetro, further ablation studies are conducted in this subsection. More exactly, we aim to investigate the effect of the ODC module, the proposed GNN modules, the multi-task architecture, and the DWA weighting method.

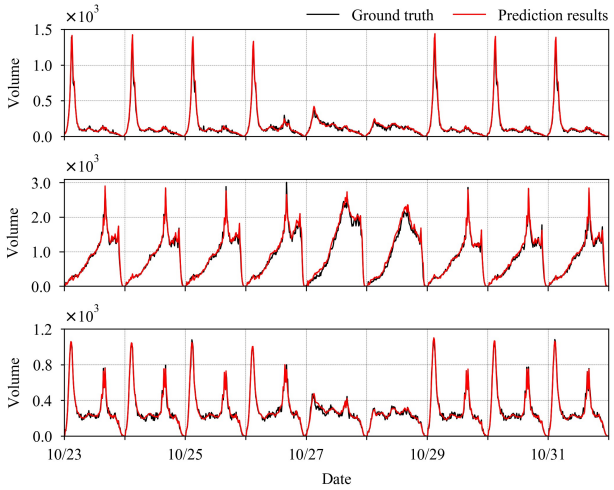


Fig.6 Ground truth and prediction results of MTLMetro on the inflows of Sihe Station (1st row), Chunxi Road Station (2nd row), and Zhongba Station (3rd row)

We first set up experiments to verify the effect of the ODC module. Three variants of MTLMetro are designed. MTLMetro-V1 takes the real-time INOD as its input. MTLMetro-V2 considers the delayed information comprising the delayed inflow and delayed INOD probability. MTLMetro-V3 utilizes the historical average INOD as the model input. The experiment results of these three variants are shown in TABLE III. Specifically, MTLMetro-V1 shows the poorest performance compared with other variants because it uses limited information as model input. By comparing MTLMetro-V2 and MTLMetro-V1, it can be seen that considering the delayed information can slightly improve the prediction performance. MTLMetro-V3 outperforms MTLMetro-V1 and -V2, demonstrating the significance of completed INOD that contains rich information for the prediction model. However, MTLMetro-V3 lacks the real-time fluctuation of passenger demand, resulting in

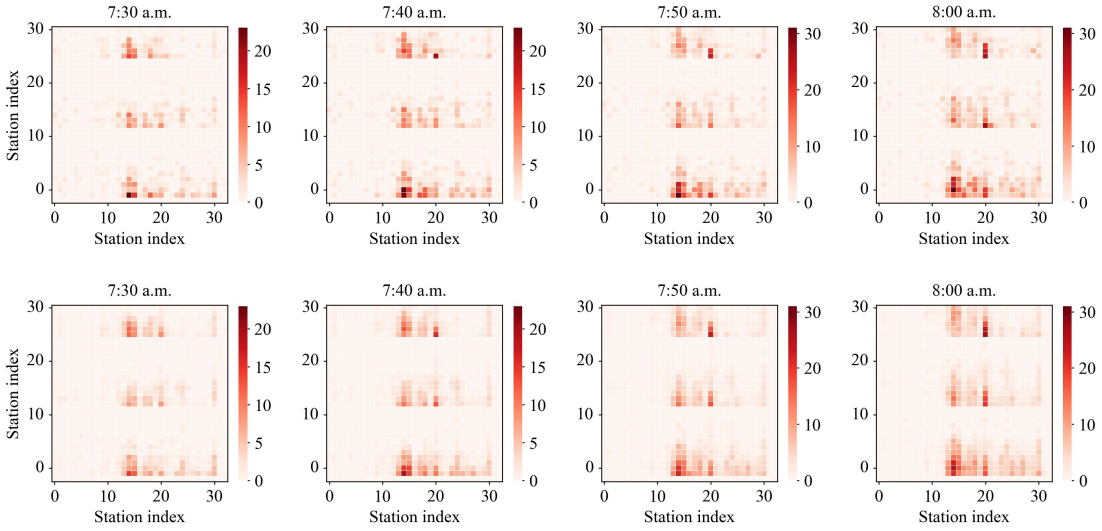


Fig.7 Ground truth (top) and prediction results (bottom) of MTLMetro on the INOD

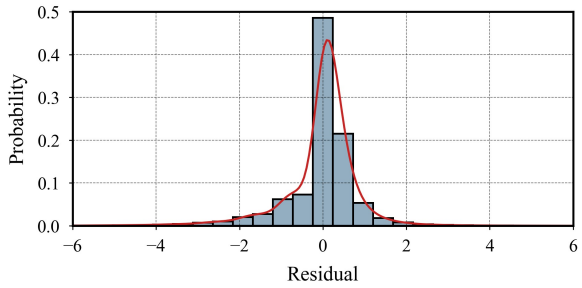


Fig.8 Residual distribution of the INOD completion results

performance degradation compared with MTLMetro. Overall, MTLMetro performs better than the other baselines because the ODC module integrates both real-time and delayed information to address the partial observability issue of INOD.

We then validate the contribution of the proposed GNN modules in MTLMetro. MTLMetro-V4 is designed by removing the PGNN module in MTLMetro. Note that, to guarantee that the MTLMetro variants can output the four demands, IGNN and OGNN are unremovable. Thus, we replace the message-passing schemes of IGNN/OGNN with those of general GCN [51] and edge-enhanced GNN (EGNN) [59], denoting them as MTLMetro-V5 and MTLMetro-V6, respectively. The prediction results are listed in TABLE III. Upon comparing MTLMetro-V4 with the original MTLMetro, a discernible improvement in the prediction of multiple demands becomes apparent, indicating that the proposed PGNN well models the relationship between inflow and outflow/OUTOD and can improve the model performance. Similar results emerge when contrasting MTLMetro-V5 and MTLMetro-V6 with the baseline MTLMetro, thereby affirming the superiority of the customized IGNN and OGNN modules. The results also shed light on the feasibility of extending the message-passing schemes in GNNs for knowledge-sharing mechanisms in the MTL paradigm.

Moreover, we explore the influence of the multi-task architecture on the final prediction results. Four single-task MTLMetro variants, MTLMetro-V7, -V8, V9, and V-10, are proposed for inflow, INOD, outflow, and OUTOD prediction, respectively. The four variants share the same model structure with MTLMetro for the corresponding subtask. The prediction results are listed in TABLE III. We can observe that the four variants are all inferior to MTLMetro in terms of three metrics. The MAE/RMSE/ MAPE increase by 1.710%/3.001%/2.630%, 2.879%/ 1.346%/9.812%, 0.806%/1.055%/1.679%, and 2.863%/ 6.294%/10.405% for inflow, INOD, outflow, and OUTOD prediction. The experiment results demonstrate the effectiveness of multi-task architecture for boosting prediction performance for all subtasks.

Finally, the ablation study about the DWA method is conducted. An MTLMetro variant without the DWA (MTLMetro-V11) is designed. MTLMetro-V11 is trained by a loss function equaling the sum of sub-tasks loss. TABLE III shows the prediction performance of MTLMetro-V11. Without balancing the training of tasks, MTLMetro-V11 shows worse performance than the proposed MTLMetro, demonstrating the advantage of the DWA method for task balancing. More details about task balancing will be discussed in the following subsection.

E. Task balancing

Training MTLMetro with the DWA weighting method leads to better prediction performance. This subsection will discuss the learning details and balancing process of multiple tasks when using DWA in MTLMetro.

First, we conduct sensitivity analysis for hyper-parameter P used to control the softness of task weights. We perform experiments on different $P \in \{1,5,10,15,20\}$. **Fig.9** shows the effect of P on the final experiment results. We can observe that the proposed model with different P outperforms MTLMetro-V11 with static and equal weights for tasks. An interesting finding is that models with a larger P tend to concentrate on the inflow and outflow tasks since they show better performance in MAE and RMSE than those with a smaller P. On the contrary, a smaller P makes the model pay more attention to the transition demands.

To explain the above phenomenon, we then detail the training process of the proposed MTLMetro. **Fig.10** demonstrates the training loss curve and dynamic weight curve for tasks when P is set as 10. At the beginning of training, the inflow and outflow tasks dominate the training with larger loss descending rates than those of the OD tasks. The reason may be that the transition demands have more complex spatial-temporal dependencies than the node

TABLE III
EXPERIMENT RESULTS OF ABLATION STUDIES

	Inflow			INOD			Outflow			OUTOD		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
MTLMetro-V1	15.907	30.591	22.739%	0.547	0.982	46.209%	18.639	31.625	27.131%	0.535	1.036	43.207%
MTLMetro-V2	15.891	30.336	22.514%	0.539	0.980	45.531%	18.631	31.600	27.052%	0.533	1.029	42.993%
MTLMetro-V3	15.851	30.003	22.463%	0.534	0.977	45.014%	18.608	31.598	27.006%	0.532	1.022	42.642%
MTLMetro-V4	15.913	30.706	22.918%	0.550	0.982	46.415%	18.646	31.657	27.222%	0.537	1.050	43.823%
MTLMetro-V5	15.917	30.754	23.000%	0.552	0.986	46.620%	18.642	31.632	27.178%	0.536	1.040	43.795%
MTLMetro-V6	15.842	29.996	22.404%	0.533	0.975	44.986%	18.680	31.700	27.257%	0.541	1.053	44.317%
MTLMetro-V7	15.884	30.239	22.553%	-	-	-	-	-	-	-	-	-
MTLMetro-V8	-	-	-	0.536	0.979	45.125%	-	-	-	-	-	-
MTLMetro-V9	-	-	-	-	-	-	18.645	31.619	27.253%	-	-	-
MTLMetro-V10	-	-	-	-	-	-	-	-	-	0.539	1.047	44.001%
MTLMetro-V11	15.681	29.399	22.157%	0.522	0.967	42.543%	18.516	31.481	26.911%	0.531	1.005	41.725%
MTLMetro	15.617	29.358	21.975%	0.521	0.966	41.093%	18.496	31.289	26.803%	0.524	0.985	39.854%

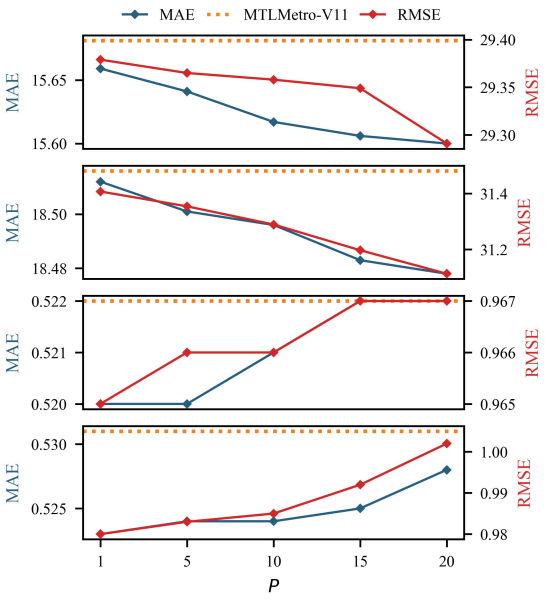


Fig.9. Parameter sensitivity analysis on inflow (1st row), outflow (2nd row), INOD (3rd row), and OUTOD (4th row)

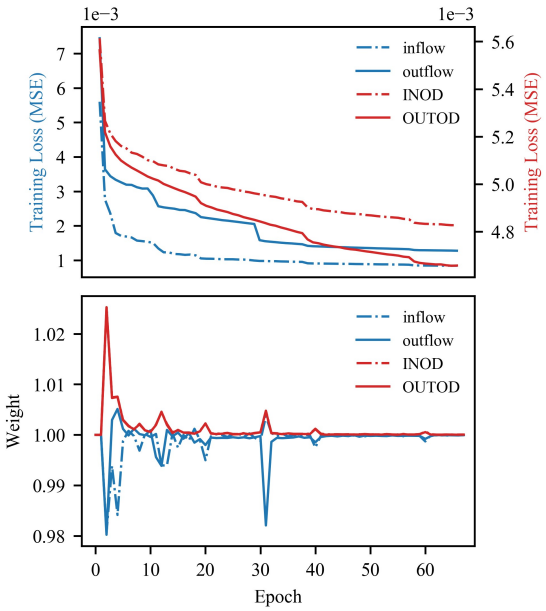


Fig.10. Training loss curve (top) and dynamic weight curve (bottom)

demands. Based on the proposed training method, smaller weights are assigned to inflow and outflow tasks to achieve task balancing. During the training, the proposed weighting method adaptively assigns weights for tasks with respect to the loss descending rate for each task. At the end of the training, equal weights are adopted since all tasks tend to converge. According to the DWA and the detailed training process, a smaller P will make relatively larger weights for OD tasks than a bigger one, and vice versa. Thus, a smaller P makes the model pay more attention to the OD tasks, resulting in better performance for OD tasks, while a larger P makes for better inflow/outflow performance.

The above experiments indicate that the priorities of MTLMetro may vary with the P value in DWA, enabling it to adapt to different prediction requirements in practical applications. For instance, when designing the train scheduling scheme, one can implement an MTLMetro model that emphasizes the OD prediction task. Additionally, it can be easily adjusted to meet the precision requirements for predicting inflow and outflow in passenger management schemes at stations. For some operation strategies in which both the node and transition demands are necessitated simultaneously, MTLMetro can ensure the overall performance with arbitrary P value, thereby contributing to the operation and management of metro systems.

VI. Conclusion

In this paper, we propose a novel deep MTL model, called MTLMetro, to jointly predict multiple demands in metro systems. Our proposed model offers several key features that differentiate it from existing methods: i) it enables end-to-end training of INOD completion and subsequent multiple demands prediction using an MTL paradigm; ii) the proposed model employs the message-passing scheme in GNNs as the knowledge-sharing mechanism in MTL to model the inherent relationships among multiple demands; iii) our model introduces a dynamic adaptive loss weighting method, DWA, to effectively balance the training of tasks. Through experiments conducted on a real dataset from the Chengdu metro system, we demonstrate the efficacy of the proposed MTLMetro model.

The empirical results highlight several noteworthy findings. Firstly, we observe that leveraging available historical and real-time information to complete the transition demand as an auxiliary task in MTLMetro provides richer information for subsequent prediction tasks, leading to improved prediction accuracy. Secondly, the knowledge-sharing mechanisms significantly influence the performance of MTL models. Our results indicate that the improper knowledge-sharing mechanisms will even degrade model performance, while mechanisms effectively capture the inherent correlations between multiple demands in IGNN, OGNN, and PGNN, leading to higher prediction precision. Thirdly, utilizing DWA to balance the training of tasks significantly enhances overall prediction performance. The adaptive tuning of task weights based on the rate of loss change for each task ensures a balanced training process. Finally, our MTLMetro model exhibits substantial improvements in prediction accuracy, as measured by MAE, RMSE, and MAPE across all passenger demand prediction tasks, outperforming benchmark models, and reinforcing the potential of our proposed model for multiple demands prediction in metro systems.

Moving forward, our future work will focus on two main aspects for model improvement. Firstly, our observation of significant differences in loss magnitudes for different sub-tasks, as illustrated in **Fig.10**, motivates future work to balance the loss magnitudes of these sub-tasks, ensuring a fair training process for each task. Secondly, we plan to extend our model to different transportation modes, such as ride-hail, taxi, and bike, and explore multiple states prediction in these domains. By addressing these aspects, we aim to further enhance the

capabilities and applicability of our model to address complex transportation demand prediction challenges.

REFERENCES

[1] H. Niu and X. Zhou, "Optimizing urban rail timetable under time-dependent demand and oversaturated conditions," *Transp. Res. Part C Emerg. Technol.*, vol. 36, pp. 212–230, 2013.

[2] N. Huan, E. Yao, and J. Zhang, "Demand-responsive passenger flow control strategies for metro networks considering service fairness and passengers' behavioural responses," *Transp. Res. Part C Emerg. Technol.*, vol. 131, p. 103335, Oct. 2021

[3] L. Li, Y. Wang, G. Zhong, J. Zhang, and B. Ran, "Short-to-medium Term Passenger Flow Forecasting for Metro Stations using a Hybrid Model," *KSCE J. Civ. Eng.*, vol. 22, no. 5, pp. 1937–1945, 2018.

[4] C. Ding, J. Duan, Y. Zhang, X. Wu, and G. Yu, "Using an ARIMA-GARCH Modeling Approach to Improve Subway Short-Term Ridership Forecasting Accounting for Dynamic Volatility," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1054–1064, 2018

[5] X. Dai *et al.*, "DeepTrend 2.0: A light-weighted multi-scale traffic prediction model using detrending," *Transp. Res. Part C Emerg. Technol.*, vol. 103, no. March, pp. 142–157, 2019

[6] J. Roos, G. Gavin, and S. Bonnevay, "A dynamic Bayesian network approach to forecast short-term urban rail passenger flows with incomplete data," *Transp. Res. Procedia*, vol. 26, pp. 53–61, 2017, doi: 10.1016/j.trpro.2017.07.008.

[7] Y. Li, X. Wang, S. Sun, X. Ma, and G. Lu, "Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks," *Transp. Res. Part C Emerg. Technol.*, vol. 77, pp. 306–328, 2017

[8] Y. Wei and M. C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," *Transp. Res. Part C Emerg. Technol.*, vol. 21, no. 1, pp. 148–162, 2012

[9] P. Noursalehi, H. N. Koutsopoulos, and J. Zhao, "Dynamic Origin-Destination Prediction in Urban Rail Systems: A Multi-Resolution Spatio-Temporal Deep Learning Approach," *IEEE Trans. Intell. Transp. Syst.*, no. January, 2021.

[10] J. Zhang, H. Che, F. Chen, W. Ma, and Z. He, "Short-term origin-destination demand prediction in urban rail transit systems: A channel-wise attentive split-convolutional neural network method," *Transp. Res. Part C Emerg. Technol.*, vol. 124, no. December 2020, p. 102928, 2021

[11] X. Ma, J. Zhang, B. Du, C. Ding, and L. Sun, "Parallel Architecture of Convolutional Bi-Directional LSTM Neural Networks for Network-Wide Metro Ridership Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2278–2288, 2019.

[12] J. Ma, J. Chan, S. Rajasegarar, G. Ristanoski, and C. Leckie, "Multi-attention 3D residual neural network for origin-destination crowd flow prediction," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2020-Novem, no. Icdm, pp. 1160–1165, 2020.

[13] Y. Zhao and Z. Ma, "Naïve Bayes-Based Transition Model for Short-Term Metro Passenger Flow Prediction under Planned Events," *Transp. Res. Rec.*, vol. 2676, no. 9, pp. 309–324, 2022.

[14] S. Hao, D. H. Lee, and D. Zhao, "Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system," *Transp. Res. Part C Emerg. Technol.*, vol. 107, no. July, pp. 287–300, 2019

[15] Y. Liu, Z. Liu, and R. Jia, "DeepPF: A deep learning based architecture for metro passenger flow prediction," *Transp. Res. Part C Emerg. Technol.*, vol. 101, no. May 2018, pp. 18–34, 2019.

[16] H. Huang, J. Mao, W. Lu, G. Hu, and L. Liu, "DEASeq2Seq: An attention based sequence to sequence model for short-term metro passenger flow prediction within decomposition-ensemble strategy," *Transp. Res. Part C Emerg. Technol.*, vol. 146, no. November 2022, p. 103965, 2023.

[17] Y. Han, S. Wang, Y. Ren, C. Wang, P. Gao, and G. Chen, "Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks," *ISPRS Int. J. Geo-Information*, vol. 8, no. 6, pp. 1–25, 2019.

[18] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, "Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7004–7014, 2020.

[19] H. Peng *et al.*, "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Inf. Sci. (Ny).*, vol. 521, pp. 277–290, 2020.

[20] J. Shi, L. Yang, J. Yang, and Z. Gao, "Service-oriented train timetabling with collaborative passenger flow control on an oversaturated metro line: An integer linear optimization approach," *Transp. Res. Part B Methodol.*, vol. 110, pp. 26–59, 2018.

[21] Y. Zhang and Q. Yang, "An overview of multi-task learning," *Natl. Sci. Rev.*, vol. 5, no. 1, pp. 30–43, 2018.

[22] H. Jia *et al.*, "ADST: Forecasting metro flow using attention-based deep spatial-temporal networks with multi-task learning," *Sensors (Switzerland)*, vol. 20, no. 16, pp. 1–23, 2020.

[23] P. Li *et al.*, "IG-Net: An Interaction Graph Network Model for Metro Passenger Flow Forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 1–11, 2023

[24] Y. Xu *et al.*, "Adaptive Feature Fusion Networks for Origin-Destination Passenger Flow Prediction in Metro Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5296–5312, 2023

[25] L. Liu, Y. Zhu, G. Li, Z. Wu, L. Bai, and L. Lin, "Online Metro Origin-Destination Prediction via Heterogeneous Information Aggregation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3574–3589, 2023

[26] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-Task Learning for Dense Prediction Tasks: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–20, 2021.

[27] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

[28] W. Jiang, Z. Ma, and H. N. Koutsopoulos, "Deep learning for short-term origin–destination passenger flow prediction under partial observability in urban railway systems," *Neural Comput. Appl.*, vol. 34, no. 6, pp. 4813–4830, 2022

[29] T. Tang, A. Fonzone, R. Liu, and C. Choudhury, "Multi-stage deep learning approaches to predict boarding behaviour of bus passengers," *Sustain. Cities Soc.*, vol. 73, no. May, p. 103111, 2021.

[30] T. Tang, R. Liu, C. Choudhury, A. Fonzone, and Y. Wang, "Predicting Hourly Boarding Demand of Bus Passengers Using Imbalanced Records From Smart-Cards: A Deep Learning Approach," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–15, 2023.

[31] J. Mao, H. Huang, W. Lu, Y. Chen, and L. Liu, "Multi-precision traffic speed predictions via modified sequence to sequence model and spatial dependency evaluation method," *Appl. Soft Comput.*, vol. 130, p. 109700, 2022.

[32] J. Mao, H. Huang, Y. Chen, W. Lu, G. Chen, and L. Liu, "Mining the Graph Representation of Traffic Speed Data for Graph Convolutional Neural Network," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2021-Septe, pp. 1205–1210, 2021.

[33] P. Xie *et al.*, "Spatio-Temporal Dynamic Graph Relation Learning for Urban Metro Flow Prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 9973–9984, 2023.

[34] Y. Zhang, K. Sun, D. Wen, D. Chen, H. Lv, and Q. Zhang, "Deep Learning for Metro Short-Term Origin-Destination Passenger Flow Forecasting Considering Section Capacity Utilization Ratio," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 7943–7960, 2023.

[35] L. Liu, J. Chen, H. Wu, J. Zhen, G. Li, and L. Lin, "Physical-Virtual Collaboration Modeling for Intra- and Inter-Station Metro Ridership Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3377–3391, 2022.

[36] D. Zhang, F. Xiao, M. Shen, and S. Zhong, "DNEAT: A novel dynamic node-edge attention network for origin-destination demand prediction," *Transp. Res. Part C Emerg. Technol.*, vol. 122, no. October 2020, p. 102851, 2021.

[37] Y. Gong, Z. Li, J. Zhang, W. Liu, and Y. Zheng, "Online Spatio-temporal Crowd Flow Distribution Prediction for Complex Metro System," *IEEE Trans. Knowl. Data Eng.*, vol. 00, no. 0, pp. 1–1, 2020.

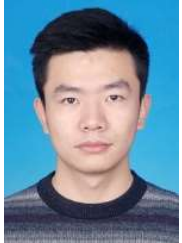
[38] Z. Cheng, M. Trépanier, and L. Sun, "Real-Time Forecasting of Metro Origin-Destination Matrices with High-Order Weighted Dynamic Mode Decomposition," *Transp. Sci.*, vol. 56, no. 4, pp. 904–918, 2022.

[39] J. Ye, F. Zheng, J. Zhao, K. Ye, and C. Xu, "Multi-View TRGRU: Transformer based Spatiotemporal Model for Short-Term Metro Origin-Destination Matrix Prediction," *arXiv Comput. Sci.*, 2021.

[40] F. Zheng, J. Zhao, J. Ye, X. Gao, K. Ye, and C. Xu, "Metro OD Matrix Prediction Based on Multi-View Passenger Flow Evolution Trend Modeling," *IEEE Trans. Big Data*, vol. 9, no. 3, pp. 991–1003, 2023.

[41] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "MMoE_Ma_2018_KDD," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1930–1939, 2018.
[42] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher, "A joint many-task model: Growing a neural network for multiple NLP tasks," *EMNLP 2017 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 1923–1933, 2017.
[43] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-Stitch Networks for Multi-task Learning," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 3994–4003, 2016.
[44] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow Prediction in Spatio-Temporal Networks Based on Multitask Deep Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, 2020.
[45] C. Zhang, F. Zhu, X. Wang, L. Sun, H. Tang, and Y. Lv, "Taxi Demand Prediction Using Parallel Multi-Task Learning Model," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–10, 2020.
[46] K. Zhang, Z. Liu, and L. Zheng, "Short-Term Prediction of Passenger Demand in Multi-Zone Level: Temporal Convolutional Neural Network with Multi-Task Learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1480–1490, 2020.
[47] S. Feng, J. Ke, H. Yang, and J. Ye, "A Multi-Task Matrix Factorized Graph Neural Network for Co-Prediction of Zone-Based and OD-Based Ride-Hailing Demand," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–13, 2021.
[48] J. Ke, S. Feng, Z. Zhu, H. Yang, and J. Ye, "Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach," *Transp. Res. Part C Emerg. Technol.*, vol. 127, no. March 2020, p. 103063, 2021.
[49] Y. Liang, G. Huang, and Z. Zhao, "Joint demand prediction for multimodal systems : A multi-task multi-relational spatiotemporal graph neural network approach," *Transp. Res. Part C*, vol. 140, no. April, p. 103731, 2022.
[50] Y. Gong, W. Liu, Z. Li, Y. Zheng, J. Zhang, and C. Kirsch, "Network-wide crowd flow prediction of Sydney trains via customized online non-negative matrix factorization," *Int. Conf. Inf. Knowl. Manag. Proc.*, vol. 1, pp. 1243–1252, 2018.
[51] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–14, 2017.
[52] P. Velicković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *arXiv*, pp. 1–12, 2017.
[53] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1724–1734, 2014.
[54] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 1871–1880, 2019.
[55] L. Zhao *et al.*, "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, 2020.
[56] Y. Zhao *et al.*, "Traffic Inflow and Outflow Forecasting by Modeling Intra- and Inter-Relationship Between Flows," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20202–20216, 2022.
[57] Y. Yang and D. Li, "NENN: Incorporate Node and Edge Features in Graph Neural Networks," *Proc. Mach. Learn. Res.*, vol. 129, no. 2017, pp. 593–608, 2020, [Online]. Available: https://proceedings.mlr.press/v129/yang20a.html.
[58] C. D. Lewis, *Industrial and business forecasting methods:a practical guide to exponential smoothing and curve fitting*. Butterworth Scientific, 1982.
[59] L. Gong and Q. Cheng, "Exploiting edge features for graph neural networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 9203–9211, 2019.

Hao Huang received the B.Sc. degree in transportation from Southwest Jiaotong University, China, in 2017. He is currently pursuing a Ph.D. degree in transportation planning and management at Southwest Jiaotong University, China. His principal research interest covers data-driven technologies for metro operation and management.



Jiannan Mao received his Master's degree in transportation engineering from Southwest Jiaotong University, China, in 2019. He is currently pursuing a Ph.D. degree in traffic engineering at Southwest Jiaotong University, China. His research interests include spatial-temporal correlation analysis and data mining.

Ronghui Liu received the B.Sc. degree from Peking University, China, and the Ph.D. degree from Cambridge University, UK. She is currently a Professor in networks and transport operations with the Institute for Transport Studies, University of Leeds. Her main research interests include developing mathematical, simulation and optimization models to analyze the dynamic and complex interplays among policy instruments, operational controls, and travellers' behavioral responses in transportation networks.

Weike Lu received the Ph.D degree in transportation from Southwest Jiaotong University in 2019. He is now an Associate Professor with the School of Rail Transportation, Soochow University, China. His primary research interests include big data analysis, Digital Twin, and network modeling.

Tianli Tang received the Ph.D. degree in transport studies from the University of Leeds in 2021. He is currently a Post-Doctoral Researcher with the School of Transportation, Southeast University. His current research focuses on dynamic management on public transport systems. He is interested in public transport planning, big data analysis.

Lan Liu received the Ph.D. degree from the Southwest Jiaotong University, China, in 2003. He is currently a Professor with the School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, China. His research interests include intelligent transportation systems and transportation information integration and fusion.