FedMetro: Efficient Metro Passenger Flow Prediction via Federated Graph Learning

Tianlong Zhang
SKLCCSE Lab
Beihang University
Beijing, China
tianlong@buaa.edu.cn

Xiaoxi He
FST
University of Macau
Macau, China
hexiaoxi@um.edu.mo

Yuxiang Wang
SKLCCSE Lab
Beihang University
Beijing, China
yuxiangwang@buaa.edu.cn

Yi Xu
SKLCCSE Lab
Beihang University
Beijing, China
xuy@buaa.edu.cn

Rendi Wu
SKLCCSE Lab
Beihang University
Beijing, China
w_rendi@buaa.edu.cn

Zhifei Wang
ICT
CARS
Beijing, China
wfbuaa@126.com

Yongxin Tong
SKLCCSE Lab
Beihang University
Beijing, China
yxtong@buaa.edu.cn

Abstract

Metro passenger flow prediction is crucial for effective urban transportation management. However, its practical adoption is hindered by data silos from distributed automatic fare collection (AFC) systems, compromising prediction accuracy. While federated graph learning facilitates privacy-preserving collaboration, existing methods struggle with the unique challenges of cross-line metro passenger flow prediction, particularly in handling time-evolving spatial correlations and heterogeneous temporal correlations. To address these challenges, we present FedMetro, a novel metro passenger flow prediction system based on federated graph learning. We introduce a federated dynamic graph learning approach with cross-attention mechanisms to capture spatial-temporal correlations in passenger flow. Additionally, we propose a dynamic mask-based communication compression method to mitigate communication bottlenecks in federated inference. Extensive evaluations on three real-world metro AFC datasets demonstrate that FedMetro significantly outperforms baseline methods, achieving up to 17.08% higher accuracy while reducing federated inference communication overhead by 77.99%. Practical deployments further confirm its effectiveness in delivering accurate station-level predictions across metro lines. Our code is available at https://github.com/AlexMufeng/FedMetro.

CCS Concepts

• **Computing methodologies → Learning paradigms**.

Keywords

Federated Learning; Metro Passenger Flow Prediction; Spatial-Temporal Graph Neural Network

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737218

ACM Reference Format:
Tianlong Zhang, Xiaoxi He, Yuxiang Wang, Yi Xu, Rendi Wu, Zhifei Wang, and Yongxin Tong. 2025. FedMetro: Efficient Metro Passenger Flow Prediction via Federated Graph Learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3711896.3737218

1 Introduction

Metro passenger flow prediction plays a crucial role in urban transportation management, enabling applications such as station congestion alerts [1], timetable optimization [2], and transportation recommendations [3]. Accurate metro passenger flow prediction relies heavily on Spatial-Temporal Graph Neural Networks (STGNNs), which leverage the spatial-temporal patterns embedded in Automatic Fare Collection (AFC) data to model complex correlations across metro networks [1, 3–8].

However, the practical implementation faces significant privacy barriers. The AFC data contains sensitive trip information protected under regulations like General Data Protection Regulation (GDPR) [9], leading to strict data isolation. As shown in Fig. 1, the Beijing metro system [10] operates 29 lines and maintains separate AFC databases for each line. These privacy constraints limit data sharing across lines, hindering effective mining and utilization of global spatial correlations within the metro network.

Federated graph learning has emerged as a promising paradigm for privacy-preserving training of STGNNs, enabling collaborative modeling of spatial-temporal correlations [11–18]. However, existing federated graph learning approaches are not directly applicable to the accurate prediction of metro passenger flow, as they face the following three significant challenges.

Time-evolving Spatial Correlations. As shown in Fig. 2, the passenger flow spatial correlations between stations are dynamic, where both topological connections and edge weights evolve over time. However, existing federated graph learning methods rely on static topologies, which can be either predefined [12–16] or data-driven [17, 18]. This limitation prevents the dynamic adaptation of spatial correlations, making it challenging to capture the evolving inter-station metro passenger flow correlations.

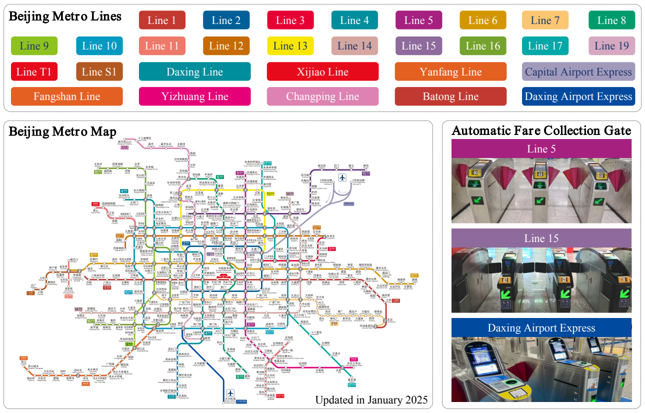


Figure 1: Each line of the Beijing metro has its own automatic fare collection system, resulting in serious data silos.

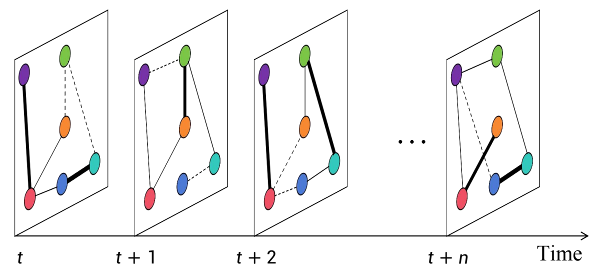


Figure 2: Time-evolving passenger flow spatial correlations across metro stations.

Heterogeneous Temporal Correlations. As shown in Fig. 3, Pingguoyuan Station experiences a clear morning peak, Zhichunlu Station has an evening peak, while Beijing Railway Station shows no significant peaks. The metro passenger flow at these three stations exhibits heterogeneous temporal correlations. Moreover, unlike traffic prediction, metro systems are segmented by lines rather than regions, which further intensifies the heterogeneous temporal correlations of passenger flow across different metro lines.

Communication Bottlenecks in Federated Inference. Metro passenger flow prediction relies on communication across lines to learn global correlations, resulting in significant overhead. Notably, during federated inference, learning these global correlations introduces a communication bottleneck, which impacts the real-time prediction needs of downstream tasks, such as congestion alerts.

In this paper, we introduce FedMetro, a novel federated metro passenger flow prediction system that enables accurate and efficient predictions in real-world distributed scenarios. We propose a federated dynamic graph learning method that employs cross-attention to generate dynamic node embeddings for each station, effectively capturing the evolving passenger flow spatial-temporal correlations without sharing raw AFC data, ensuring compliance with privacy regulations. This method effectively models the global metro passenger flow correlations, overcoming the challenges of time-evolving spatial correlations and the heterogeneity of temporal correlations, thereby improving the accuracy of metro passenger

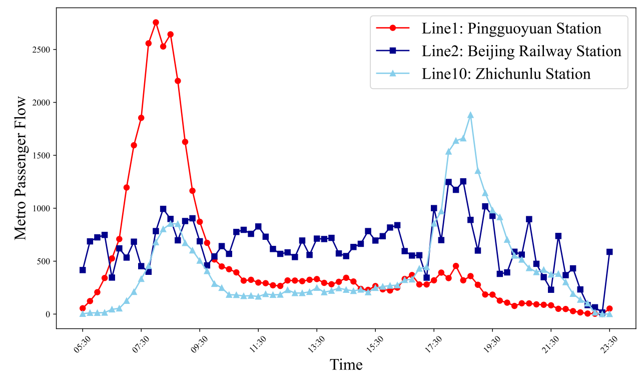


Figure 3: Heterogeneous temporal correlations of metro passenger flow between three different stations.

flow predictions. Furthermore, we significantly reduce communication overhead by dynamically sparsifying the spatial correlation graph, while maintaining high prediction accuracy and effectively addressing communication bottlenecks in federated inference.

The main contributions of the paper are as follows:

- To the best of our knowledge, this is the first research to apply federated graph learning for metro passenger flow prediction, addressing privacy constraints in the distributed AFC systems of large urban metro networks.
- We introduce FedMetro, a novel system that adapts to the time-evolving spatial correlations and heterogeneous temporal correlations of metro passenger flow, while overcoming communication bottlenecks in federated inference.
- Evaluations show that FedMetro reduces communication overhead by 77.99% and outperforms state-of-the-art methods in accuracy. Deployment results demonstrate its exceptional prediction performance for stations with heterogeneous temporal correlations in real-world applications.

In the rest of this paper, we first introduce the datasets and problem definitions in Sec. 2. Next, we present our proposed solution in Sec. 3. We then conduct extensive experiments in Sec. 4 and discuss the practical deployment in Sec. 5. Finally, we review related works in Sec. 6 and conclude in Sec. 7.

2 Preliminaries

2.1 Metro Automatic Fare Collection (AFC) Data

Metro companies utilize Automatic Fare Collection (AFC) systems to streamline fare transactions and track real-time passenger flow data [5]. These AFC systems generate records when passengers tap smart cards or scan QR codes at station gates for entry or exit events. Each AFC record includes a user ID, timestamp, event type (entry or exit), metro line name, and metro station name. Tab. 1 provides several examples of AFC records. However, in accordance with data protection regulations like the GDPR [9], AFC data for each metro line is stored independently. At transfer stations, the data are jointly managed by the operating companies of the involved lines. To facilitate prediction tasks, metro operation times are commonly divided into fixed-length intervals (*e.g.* 15-minute segments), with

Table 1: Examples of AFC records.

User ID	Timestamp	Event Type	Line	Station
BIAEFDCBI	2017/06/03 20:14:30	entry	L1/L10	GuoMao
BCHIDEBCC	2017/06/04 09:12:10	entry	L1	SiHui
CFCBFCGGC	2017/06/05 20:31:25	exit	L5/L6	DongSi

passenger flow counts within each interval [3]. Our AFC datasets were collected from three major cities in China: Beijing, Shanghai, and Hangzhou, whose metro systems are highly representative of large urban metro networks.

2.2 Graph Modeling of Federated Metro Passenger Flow

The metro AFC data is represented as a chronological sequence $\boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^T$, where each signal graph $\boldsymbol{X}^t \in \mathbb{R}^{N\times F}$ captures the observations at the t-th timestep. Here, N represents the number of stations and F represents the feature dimension (*e.g.* passenger flow). Due to privacy constraints, Metro AFC data is stored independently by each metro line. Consider M metro lines, denoted as $C = C_1, C_2, ..., C_M$. Each metro line C_i maintains a local dataset $\mathcal{D}_i = \{\mathcal{G}_i, \boldsymbol{X}_i^{1:T}\}$, where $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i\}$ represents the local spatial correlations graph with N_i stations. The global graph $\mathcal{G} = \bigcup_{i=1}^{M} \mathcal{G}_i$ is constructed by combining all local graphs, with the total number of stations given by $N = \sum_{i=1}^{M} N_i$.

Signal Graph. The signal graphs $\boldsymbol{X}^{1:T} \in \mathbb{R}^{T\times N\times F}$ are naturally partitioned across the metro lines as:

$$\boldsymbol{X}^{1:T} = \begin{bmatrix} \boldsymbol{X}_1^1 & \cdots & \boldsymbol{X}_1^T \\ \vdots & \ddots & \vdots \\ \boldsymbol{X}_M^1 & \cdots & \boldsymbol{X}_M^T \end{bmatrix} \tag{1}$$

where $\boldsymbol{X}_i^{1:T} \in \mathbb{R}^{T\times N_i\times F}$ is the local signal graph for the N_i nodes over T time intervals. This partitioned representation naturally aligns with the administrative divisions of metro operations, where each line controls its infrastructure and data.

Spatial Correlations Graph. The graph $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i\}$ represents the passenger flow correlations network of the metro system, with nodes $\mathcal{V}$ ($|\mathcal{V}| = N$) and edges $\mathcal{E}$ defining spatial correlations between stations. The adjacency matrix $\boldsymbol{A}$ encodes global spatial correlations and is crucial for accurate modeling:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \cdots & \boldsymbol{A}_{1M} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}_{M1} & \cdots & \boldsymbol{A}_{MM} \end{bmatrix}, \tag{2}$$

where $\boldsymbol{A}_{ij} \in \mathbb{R}^{N_i\times N_j}$ corresponds to the correlation between stations in $\mathcal{V}_i$ and $\mathcal{V}_j$. The matrix $\boldsymbol{A}$ is typically constructed using one of the following approaches:

- A predefined physical connectivity graph based on the metro structural topology [3];
- Multiple predefined graphs integrating the metro structural topology with additional factors such as weather, holidays, and points of interest (POIs) [5];
- Data-driven topology learning, where station passenger flow correlations are learned from the AFC data [7, 8, 19].

Among these, data-driven topology learning is considered state-of-the-art because it allows the model to capture the spatial correlations of passenger flow between stations more comprehensively, making it particularly suitable for large metro systems.

2.3 Federated Metro Passenger Flow Prediction

We propose a client-server federated learning framework where M metro lines operate as distributed clients, collaboratively optimizing a predictive model through coordinated aggregation by a central server [20]. This framework allows for the decentralized training of a model that captures metro passenger flow dynamics while preserving the privacy of each line's data.

Problem Definition. Given a current prediction timestep $t \in \{\tau - T_{\text{in}} + 1, \cdots, \tau\}$, we aim to learn a predictive function $\mathcal{F}(\cdot)$ that maps a sequence of T_{in} historical observations to the next T_{out} future observations:

$$\hat{X}^{(\tau+1):(\tau+T_{out})} \leftarrow \mathcal{F}(X^{(\tau-T_{in}+1):\tau}) \tag{3}$$

Privacy Constraint. To ensure data privacy, the server does not have access to the local datasets $\mathcal{D}_i$ of any client. Additionally, each client, C_i, is prohibited from sharing its own graph signals $\boldsymbol{X}_i^{1:T}$ or its local graph structure $\mathcal{G}_i$ with other clients, ensuring that sensitive passenger flow data and station-specific patterns remain confidential. This privacy constraint supports collaborative learning without the need for centralized data collection, maintaining the operational independence of different metro lines.

3 FedMetro System

3.1 System Overview

To achieve efficient federated metro passenger flow prediction, we propose a novel system named FedMetro. As shown in Fig. 4, we use a client-server federated learning framework[20, 21]. The server is responsible for global correlations and training parameters aggregation, while the client consists of three modules: ① Dynamic Embedding and Mask Generation; ② Communication Compression; ③ Correlations Recovery and Training.

Specifically, we use cross-attention to capture the passenger flow temporal correlations for each station by matching current observations $\boldsymbol{X}_i^t$ with historical patterns $\boldsymbol{X}_i^{1:t}$. Then, we model the dynamic global spatial correlations between stations across metro lines by constructing an embedding $\boldsymbol{E}_i^t$ for each metro line at each timestamp. Next, we propose a communication compression method. We sparsify $\boldsymbol{H}_i^t$ with dynamic masks M_i^t derived from historical patterns, then aggregate $\boldsymbol{E}_i^{t\top}$ to obtain the compressed result $\boldsymbol{AGG}_i^t$, reducing communication overhead. After that, each client uploads the local correlations $\boldsymbol{AGG}_i^t$ to the server. The server aggregates the global correlations and returns it to the clients. Each client then performs dimensional recovery and proceeds with local spatial-temporal graph convolution training. Finally, after the local training rounds, each client sends training parameters to the server for aggregation and updating of global parameters.

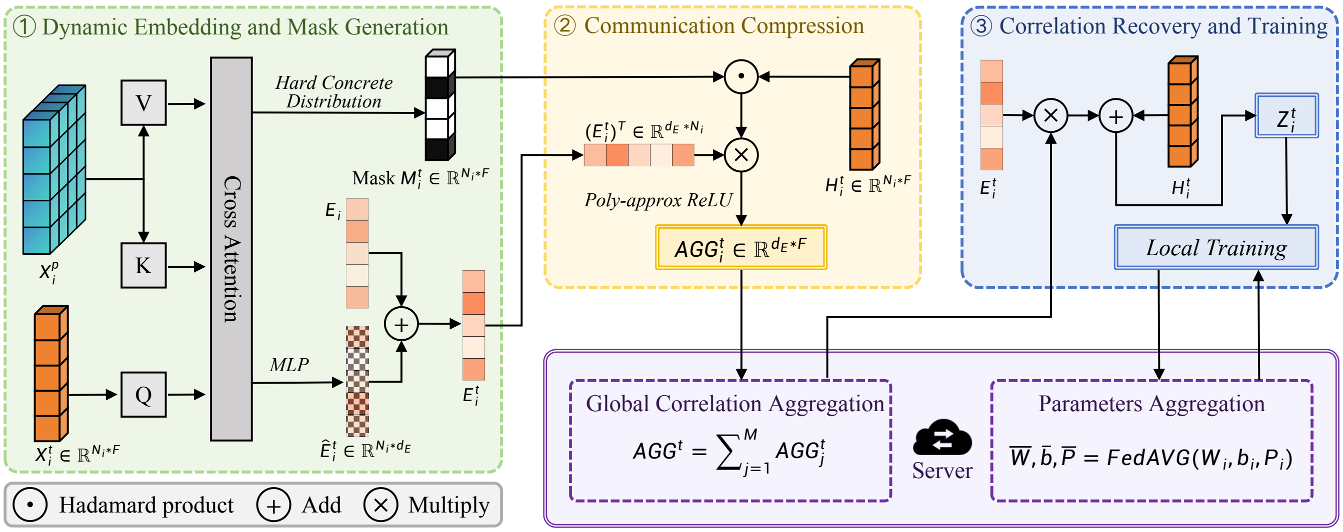


Figure 4: An overview of FedMetro system.

3.2 Dynamic Embedding and Mask Generation

Metro passenger flow prediction faces challenges due to time-evolving spatial correlations and heterogeneous temporal correlations. To address this, we use cross-attention that integrates multi-scale information from the historical time dimension, enabling more effective modeling of dynamic spatial-temporal correlations.

Historical Pattern Mining. To efficiently represent the historical passenger flow data, we apply down-sampling and patching techniques to compress the data $X_i^{1:t}$ into X_i^p. For each station's passenger flow data, we first perform 1D average pooling along the temporal dimension with kernel size k and stride k. The down-sampled data is then divided into p non-overlapping patches, which reduces the number of tokens.

For the current timestamp t, we use cross-attention to capture the pattern correlations between the current observation X_i^t of client i and the historical patch data X_i^p:

$$Q = X_i^t W_{i,Q}, \qquad K = V = X_i^p W_{i,K} + e_{pos} \tag{4}$$

where $W_{i,Q}$ and $W_{i,K}$ are learnable parameter matrices, and e_{pos} is the position embedding. We then apply cross-attention along the temporal dimension to mine the pattern dependencies:

$$D_i^t = CrossAttention(Q, K, V) = Softmax(\frac{QK^\top}{\sqrt{d}})V \tag{5}$$

D_i^t is used to generate both dynamic embedding and mask later.

Dynamic Embedding Generation. As mentioned in Sec. 2.2, data-driven topology learning methods are ideal for constructing the spatial correlations graph A:

$$A = softmax\left(ReLU\left(E \cdot E^\top\right)\right) \tag{6}$$

where each row of $E \in \mathbb{R}^{N \times d_E}$ represents the learnable embedding of a metro station. The Adaptive Graph Convolution Networks (AGCN) [7] incorporate this data-driven topology learning approach into the GCN layer:

$$H^{(l)} = \sigma\left(\left(I_N + ReLU\left(E \cdot E^\top\right)\right) \cdot H^{(l-1)} \cdot E \cdot W + E \cdot b\right) \tag{7}$$

In this context, $H^{(l-1)} \in \mathbb{R}^{N \times F^{(l-1)}}$ and $H^{(l)} \in \mathbb{R}^{N \times F^{(l)}}$ represent the input and output features of the l-th layer. I_N denotes the identity matrix, and $\sigma(\cdot)$ is the nonlinear activation. $W \in \mathbb{R}^{d \times F^{(l-1)} \times F^{(l)}}$ and $b \in \mathbb{R}^{d \times F^{(l)}}$ represent the trainable weight and bias pool. Through the computation of the dot product between E and W, a personalized set of parameters is learned for each station to adaptively capture the spatial correlations.

However, the AGCN learns only a static spatial correlations graph during training, which does not account for the rapidly changing metro passenger flow. To address this issue, we utilize cross-attention to construct a dynamic spatial correlations graph by learning a node embedding E_i^t for each timestamp in client i:

$$\hat{E}_i^t = D_i^t W_{i,E}, \qquad E_i^t = E_i + \hat{E}_i^t \tag{8}$$

where $\hat{E}_i^t$ is the time increment embedding, which captures the temporal changes of E at each timestep. $W_{i,E} \in \mathbb{R}^{d_D \times d_E}$ represents the learnable model parameters.

Dynamic Mask Generation. For the subsequent communication compression, we generate a dynamic mask matrix M_i^t for each client i using the D_i^t obtained from cross-attention:

$$\begin{aligned} &O_i^t = D_i^t W_{i,O}, \qquad z \sim \mathcal{U}(0,1), \\ &s = sigmoid\left(\log z - \log(1-z) + \log(o)\right)/\beta, \\ &\hat{s} = s(\zeta - \gamma) + \gamma, \qquad m = \min\left(1, \max(0, \hat{s})\right) \end{aligned} \tag{9}$$

where $W_{i,O} \in \mathbb{R}^{d_D \times F}$ denotes a trainable parameter matrix. z is sampled from a uniform distribution. To address the non-differentiability of the Bernoulli distribution, we adopt the hard concrete distribution as a continuous relaxation [22, 23]. $s, \hat{s}, o, m$ are the elements of the matrices $s, \hat{s}, W_{i,O}, M_i^t \in \mathbb{R}^{N_i \times F}$. β is the temperature value and $(\zeta - \gamma)$ is the interval.

Algorithm 1: Dynamic Embedding and Mask Generation

input : Data sequence $X_i^{1:t}$ on C_i;
Personalized node embeddings E_i on C_i;
output: Dynamic embedding E_i^t and mask M_i^t on C_i;
Compress $X_i^{1:t}$ into historical patch data X_i^p;
//Historical Pattern Mining according to Eq. (4) and Eq. (5)
$D_i^t \leftarrow CrossAttention(X_i^t, X_i^p)$;
//Dynamic Embedding Generation according to Eq. (8)
$E_i^t \leftarrow E_i + D_i^t W_{i,E}$;
//Dynamic Mask Generation according to Eq. (9)
$O_i^t \leftarrow D_i^t W_{i,O}$;
$M_i^t \leftarrow HardConcrete(O_i^t)$;
return (E_i^t, M_i^t);

This subsection introduces the Dynamic Embedding and Mask Generation module in FedMetro. As shown in Algorithm 1, both the embedding and the mask are derived from the dynamic representation D_i^t, which is learned through a cross-attention mechanism. For each metro station of client C_i, the node embeddings are personalized to address the heterogeneous temporal correlations across different stations. Moreover, by leveraging a dynamic graph, both the embedding and the mask evolve over time at each timestep t, enabling the model to effectively capture the time-varying spatial correlations in metro passenger flow prediction.

3.3 Global Spatial Correlations Learning

Spatial Correlation Modeling. Inspired by FedGTP [18], we extend AGCN to the federated learning setting:

$$H_i^{(l)} = \sigma\left(Z_i^t(H^{(l-1)}) \cdot E_i^t \cdot W_i + E_i^t \cdot b_i\right) \tag{10}$$

where Z_i^t denotes the spatial correlations in client i. For brevity, we omit the layer ID:

$$Z_i^t(H) = H_i + \sum_{j=1}^{M}\left(ReLU\left(E_i^t \cdot (E_j^t)^\top\right) \cdot H_j\right) \tag{11}$$

Here, both $E_i^t \in \mathbb{R}^{N_i \times d_E}$ and $H_i \in \mathbb{R}^{N_i \times F}$ need to be uploaded. However, $H_i^{(0)} = X_i^t$ violates the privacy constraints of the original signal graph. Furthermore, in practical deployments, the large number of stations N_i for each metro line hinders scalability. Therefore, we next introduce a spatial correlations compression method.

Global Spatial Correlation Aggregation. To efficiently construct the global spatial correlations Z_i while adhering to the privacy constraints, we redesign its computation to minimize the size of intermediate results uploaded to the server:

$$Z_i^t(H) = H_i + ReLU\left(E_i^t\right) \cdot \sum_{j=1}^{M}\left(\left(ReLU\left(E_i^t\right)^\top \cdot H_i\right)\right) \tag{12}$$

However, directly decomposing *ReLU* limits the model's ability to capture complex nonlinear relationships. Studies [18, 24] show that *Poly-approx ReLU* with polynomials via Taylor expansion reduces training errors. Therefore, we use a K-order polynomial function to approximate the activation and preserve non-linearity:

$$\mathcal{P}_K(x) = \sum_{k=0}^{K} p_k x^k \tag{13}$$

Therefore, the spatial correlation $Z_i(H)$ is represented as:

$$\begin{aligned} Z_i^t(H) &= H_i + \sum_{j=1}^{M}\left(\sum_{k=0}^{K}\left(p_k \cdot \left(E_i^t \cdot (E_j^t)^\top\right)^k\right) \cdot H_j\right) \\ &= H_i + \sum_{k=0}^{K} p_k \cdot \mathcal{F}_K(E_i^t) \cdot \sum_{j=1}^{M}\left(\left(\mathcal{F}_K\left(E_j^t\right)\right)^\top \cdot H_j\right) \end{aligned} \tag{14}$$

Here, $\mathcal{F}_K(\cdot)$ decomposes the spatial correlations graph without information loss by applying the k-th Cartesian power to each row.

To reduce communication overhead, we sparsify H_i^t with the mask matrix M_i^t, retaining core values and filtering out noise to compress the intermediate results uploaded to the server:

$$AGG_i^t = \left(\mathcal{F}_K\left(E_i^t\right)\right)^\top \cdot \left(H_i \odot M_i^t\right) \tag{15}$$

Ultimately, client i uploads the intermediate result $AGG_i^t \in \mathbb{R}^{d_E^K \times F}$ to the server. The server aggregates the AGG_i^t to obtain the global aggregation result AGG^t:

$$AGG^t = \sum_{i=1}^{M}\left(AGG_i^t\right) \tag{16}$$

Communication Analysis. We conduct a theoretical analysis of the total communication overhead for each client uploading intermediate results to the server at time t. In the unoptimized AGCN global spatial correlation learning process, the total communication cost is $N \times d_E^K + N \times F$. After optimization, the communication overhead is reduced to $(d_E^K \times F)/S$, where S is the average compression ratio of AGG_i^t. The dimension F is much smaller than the number of nodes N. Our method is particularly effective for metro systems in large cities like Beijing and Shanghai, where the number of stations N is large, improving the scalability of federated graph learning for passenger flow prediction.

3.4 Federated Training and Inference

Spatial-Temporal Fusion. We combine the restructured global spatial correlation with temporal modeling to capture time-evolving metro passenger flow patterns. Specifically, we replace the MLP layer in GRU with the global spatial correlation:

$$Z_i^t(H) = H_i + \sum_{k=0}^{K} p_k \cdot \mathcal{F}_K(E_i^t) \cdot AGG^t \tag{17}$$

At the timestep t, the federated training process of FedMetro is defined as:

$$\begin{aligned} z_i^t &= \sigma\left(Z_i[X_i^t \,||H^{t-1}] \cdot E_i^t \cdot W_{i,z} + E_i^t \cdot b_{i,z}\right), \\ r_i^t &= \sigma\left(Z_i[X_i^t \,||H^{t-1}] \cdot E_i^t \cdot W_{i,r} + E_i^t \cdot b_{i,r}\right), \\ \tilde{H}_i^t &= tanh\left(Z_i[X_i^t \,||(r^t \odot H^{t-1})] \cdot E_i^t \cdot W_{i,\tilde{H}} + E_i^t \cdot b_{i,\tilde{H}}\right), \\ H_i^t &= z_i^t \odot H_i^{t-1} + (1 - z_i^t) \odot \tilde{H}_i^t \end{aligned} \tag{18}$$

All learnable parameters in Eq. (18) can be optimized end-to-end through backpropagation through time. During local training, each client C_i performs multiple training iterations according to Eq. (18). Afterward, each client C_i uploads the weight pool W_i and bias pool b_i to server for parameter aggregation [25], resulting in the global parameters $\overline{W}$, $\overline{b}$ and $\overline{P}$.

Training Objectives. Due to significant spatial differences between metro lines, which lead to notable variations in passenger flow, there exist substantial non-IID(independent and identically distributed) regional characteristics. Therefore, we formulate the personalized federated learning objective [26]:

$$\mathcal{L}_{FedMetro} = \sum_{i=1}^{M} \sum_{t=1}^{T} \frac{N_i}{N} \left(\mathcal{L} \left(\mathcal{F}, \Theta_i, \boldsymbol{H}_i^t \odot \boldsymbol{M}_i^t \right) + \lambda \left\| AGG_i^t \right\|_0 \right) \tag{19}$$

where $\mathcal{L}$ denotes the federated loss function of model $\mathcal{F}$ with learnable parameters Θ. λ represents the regularization weight for the L_0-norm of AGG_i^t to enforce communication compression.

Federated Inference. During inference, the server only assists each client C_i in learning the global spatial correlation as Eq. (17). Our compression using the mask matrix M_i^t effectively reduce the communication overhead of the local spatial correlation AGG_i^t uploaded by each client C_i to the server. Moreover, in practical deployment, we set a threshold for the communication compression ratio to balance the trade-off between prediction accuracy and communication overhead. This effectively solves the communication bottleneck in federated inference and allows more time for subsequent station congestion alerts and emergency measures.

3.5 System Implementation

The previous subsections presented introduce the main modules of FedMetro, which enable efficient metro passenger flow prediction via dynamic graphs and mask mechanisms under a federated framework. We now detail the design and implementation of the FedMetro system, as outlined in Algorithm 2. In this system, M metro lines operate as distributed clients that collaboratively optimize a predictive model under the coordination of a central server. In each training round, client C_i performs local learning of dynamic node embeddings and corresponding mask matrices according to Algorithm 1. The node embeddings are then transformed via *Poly-approx ReLU*, and communication compression is applied to derive $\boldsymbol{AGG}_i^t$. These compressed representations are aggregated at the server to compute the global spatial correlations $\boldsymbol{AGG}^t$, which are broadcast back to all clients. Each client uses this global information to perform local spatial-temporal graph convolution training, as defined in Eq. (18). After several local training, each client uploads its model parameters. Finally, The server applies the FedAVG [25] to aggregate these parameters and update the global model.

Algorithm 2: FedMetro Framework

input : Data sequence $\boldsymbol{X}_i^{1:t}$ for each C_i;
The number of global and local rounds R_g, R_l;
output: Trained model weights $(\boldsymbol{W}_i, \boldsymbol{b}_i, \boldsymbol{P}_i, \boldsymbol{E}_i)$ for each C_i;

Initialize global model weights with $(\boldsymbol{W}_g^{(0)}, \boldsymbol{b}_g^{(0)}, \boldsymbol{P}_g^{(0)})$;
Initialize personalized node embeddings with $\{\boldsymbol{E}_i^{(0)}\}_{i=1}^{M}$;
for *global round* ***in*** *global rounds* R_g **do**
  for *each client* C_i ***in*** *clients* C_M **do**
    Receives global model weights from server to update $(\boldsymbol{W}_i, \boldsymbol{b}_i, \boldsymbol{P}_i)$;
    for *local round* ***in*** *local rounds* R_l **do**
      for *timestep t* ***in*** *timesteps* **do**
        //Dynamic Embedding and Mask Generation
        $(\boldsymbol{E}_i^t, \boldsymbol{M}_i^t) \leftarrow$ Algorithm 1 $(\boldsymbol{X}_i^{1:t}, \boldsymbol{E}_i)$;
        //Poly-approx ReLU according to Eq. (8)
        $\mathcal{F}_K(\boldsymbol{E}_i^t) \leftarrow \left\{ f_k(\boldsymbol{E}_i^t) \right\}_{k=0}^{K}$;
        //Communication Compression
        $\boldsymbol{AGG}_i^t \leftarrow \mathcal{F}_K^\top(\boldsymbol{E}_i^t) \cdot (\boldsymbol{H}_i^t \odot \boldsymbol{M}_i^t)$;
        //server: Global Correlation Aggregation
        $\boldsymbol{AGG}^t \leftarrow SecureSummation(\{\boldsymbol{AGG}_i^t\}_{i=1}^{M})$;
        //Correlation Recovery and Training
        Forwards spatial-temporal correlation modeling according to Eq. (18);
        Update $(\boldsymbol{W}_i, \boldsymbol{b}_i, \boldsymbol{P}_i, \boldsymbol{E}_i)$ through gradient descent.
    Sends $(\boldsymbol{W}_i, \boldsymbol{b}_i, \boldsymbol{P}_i)$ to server;
  //server: Parameters Aggregation
  $(\overline{\boldsymbol{W}}, \overline{\boldsymbol{b}}, \overline{\boldsymbol{P}}) \leftarrow FedAvg(\{\boldsymbol{W}_i^{(0)}, \boldsymbol{b}_i^{(0)}, \boldsymbol{P}_i^{(0)}\}_{i=1}^{M})$;
return $(\boldsymbol{W}_i, \boldsymbol{b}_i, \boldsymbol{P}_i, \boldsymbol{E}_i)$ for each C_i;

Table 2: Statistics of datasets.

	BJMetro	SHMetro	HZMetro
City	Beijing, China	Shanghai, China	Hangzhou, China
Lines	15	14	3
Stations	276	288	80
Edges	906	958	248
Period	2016/02/29–2016/04/02	2016/07/01–2016/09/30	2019/01/01–2019/01/25

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate FedMetro on three real-world metro datasets collected from Beijing (BJMetro), Shanghai (SHMetro), and Hangzhou (HZMetro). Following realistic deployment scenarios, each dataset is partitioned according to metro line, containing both inflow and outflow records aggregated at 15-minute intervals. The dataset statistics are summarized in Tab. 2.

- **BJMetro**: Collected from Beijing metro AFC systems between February 29 and April 2, 2016. The 2016 Beijing metro network comprised 15 lines serving 276 stations.
- **SHMetro**: Collected from Shanghai metro AFC systems. This dataset contains 811.8 million transaction records collected from July 1 to September 30, 2016. During this period, 288 stations were actively operational.
- **HZMetro**: Collected from Hangzhou metro AFC system between January 1-25, 2019. This dataset represents the metro network with 80 stations and 248 physical edges.

Baselines. We establish two baseline categories to evaluate the performance of FedMetro. The first category of baselines comprises local GCN-based methods for predicting metro passenger flow.

Table 3: Performance comparison of our method and baselines.

Method	BJMetro						SHMetro						HZMetro					
	Inflow			Outflow			Inflow			Outflow			Inflow			Outflow		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Graph-WaveNet	46.73	89.35	41.58%	45.16	82.61	59.93%	34.02	84.90	51.79%	33.07	86.03	49.38%	33.79	70.34	37.85%	31.66	68.15	44.51%
PVCGN	55.16	120.97	29.50%	56.87	137.75	29.36%	30.44	67.88	38.68%	30.02	77.32	34.61%	33.10	65.29	36.17%	30.23	65.66	29.41%
STDGRL	37.98	75.89	28.22%	39.34	<u>73.17</u>	<u>27.66%</u>	30.10	62.43	<u>25.44%</u>	30.88	75.88	<u>28.30%</u>	31.76	57.92	22.54%	33.33	71.40	25.21%
MFVSTGNN	39.04	82.77	36.54%	38.60	104.49	74.17%	<u>28.55</u>	<u>60.47</u>	29.66%	<u>29.19</u>	80.93	42.79%	28.22	<u>47.44</u>	40.50%	29.34	60.39	40.59%
FedGTP	<u>31.87</u>	<u>62.11</u>	<u>25.38%</u>	<u>36.31</u>	83.55	27.69%	32.16	67.32	26.59%	31.98	<u>74.15</u>	28.77%	<u>28.02</u>	49.80	<u>20.41%</u>	<u>28.41</u>	<u>56.39</u>	<u>24.51%</u>
FedMetro (Ours)	**29.70**	**58.84**	**21.73%**	**30.11**	**65.59**	**23.15%**	**27.97**	**56.15**	**23.75%**	**29.13**	**68.56**	**26.82%**	**26.21**	**46.53**	**19.46%**	**27.36**	**54.42**	**22.67%**
Improvement	6.81%	5.26%	14.38%	17.08%	10.36%	16.31%	2.03%	7.14%	6.64%	0.21%	7.54%	5.23%	6.46%	1.92%	4.65%	3.70%	3.49%	7.51%

- Graph WaveNet [19]: A spatial-temporal model that captures spatial correlations via adaptive adjacency matrices constructed from learnable node embeddings.
- PVCGN [5]: An encoder-decoder model with dual-branch gated recurrent units (GC-GRU and FC-GRU), where the GC-GRU captures local station dependencies and FC-GRU models global metro dynamics.
- STDGRL [8]: A state-of-the-art approach for local metro passenger flow prediction based on dynamic graph to capture the distinct patterns of different stations

The second category of baselines includes federated graph learning-based methods for traffic prediction.

- MFVSTGNN [27]: A federated graph learning framework with client-specific graph representation learning.
- FedGTP [18]: A state-of-the-art federated graph learning-based traffic prediction method that fully utilizes inter-client spatial dependencies for accurate predictions.

We conducted all the experiments under the same setup. To meet the requirements of practical deployment applications, we used the previous 6 intervals to predict the next 6 intervals.

Metrics. We adopt three widely-used metrics in our experiments: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). Lower values of these metrics reflect better prediction performance.

Experimental Environment. All experiments are implemented in PyTorch 1.13.1 on Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz and four NVIDIA A100 GPUs with 40GB memory.

4.2 Performance Comparison with Baselines

Tab. 3 presents the overall performance of our method and baselines on three different datasets. The best results for each metric are highlighted in bold, and the second-best results are underlined. We observe that FedMetro outperforms all baselines across all evaluation metrics, with metro passenger flow prediction accuracy improving by up to 17.08%. This improvement showcases the effectiveness and robustness of our approach in handling complex metro systems. Additionally, the results show that federated graph learning methods outperform local GCN-based approaches in most cases. This performance gap can be attributed to the inherent limitations of local GCN-based methods, which are constrained to making flow predictions for individual metro lines due to privacy concerns in real-world applications. In contrast, federated learning enables

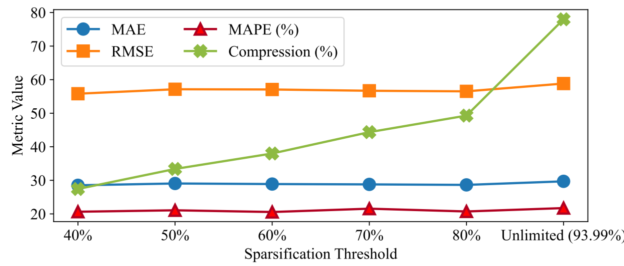


Figure 5: The results of communication compression study.

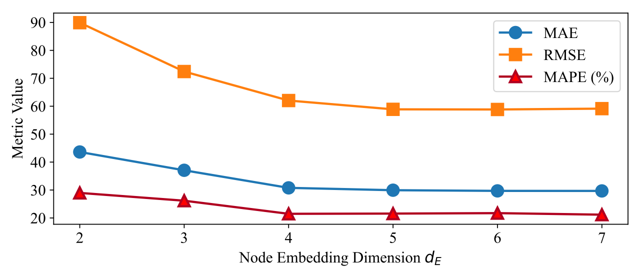


Figure 6: The results of node embedding dimension study.

cross-line prediction by leveraging shared knowledge across multiple metro lines while preserving data privacy. This capability is crucial for accurate and privacy-preserving metro passenger flow prediction, as it allows for a more holistic understanding of passenger movement patterns across the entire metro network.

4.3 Communication Compression Study

To explore the trade-off between precision and compression in FedMetro, we introduce a sparsification threshold to control the sparsity level of the intermediate results $AGG_i^{(\tau)}$ uploaded by each client C_i to the server. Based on the experimental results from BJMetro, shown in Fig. 5, we observe that increasing the sparsification threshold improves the communication compression ratio, with only a minimal decline in prediction accuracy. Specifically, when the threshold is unlimited, the sparsity of the AGG_i^{τ} uploaded by each client C_i can reach up to 93.99%, resulting in a communication compression ratio of 77.99%. As illustrated in Tab. 3, even at such a high compression level, our model still significantly outperforms all baseline models in terms of prediction accuracy.

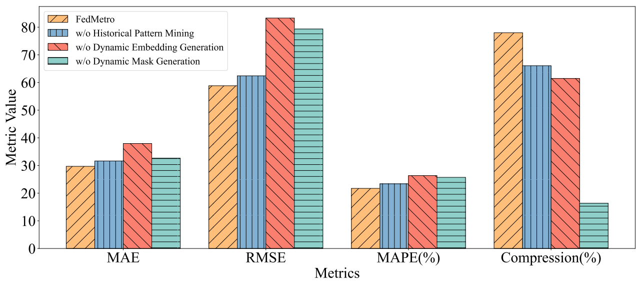


Figure 7: Ablation study on BJMetro.

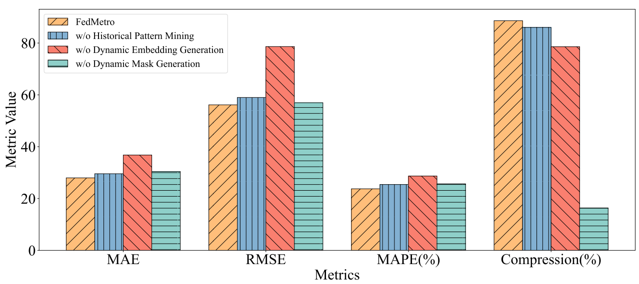


Figure 8: Ablation study on SHMetro.

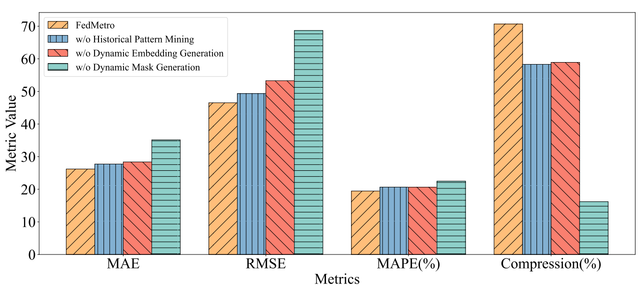


Figure 9: Ablation study on HZMetro.

4.4 Node Embedding Dimension Study

The node embedding dimension d_E is a key hyperparameter that directly affects prediction accuracy and model capacity. We evaluate its impact on the BJMetro dataset, as shown in Fig. 6. The findings reveal that when $d_E = 2$, prediction accuracy is suboptimal. However, as d_E increases to 4, there is a noticeable improvement in prediction accuracy. We attribute this improvement to the personalized nature of passenger flow at different metro stations. As d_E increases further, the improvement in prediction accuracy becomes marginal, indicating that the model has reached an optimal balance between accuracy and capacity.

4.5 Ablation Study

To clearly isolate the contribution of each component in FedMetro, we conduct ablation studies on the Dynamic Embedding Generation, Dynamic Mask Generation, and Historical Pattern Mining modules. Fig. 7, Fig. 8 and Fig. 9 reveal that these modules are complementary and indispensable. Removing Dynamic Embedding Generation causes the most severe degradation across MAE, RMSE, and MAPE, highlighting the need for dynamic node embeddings to capture time-evolving spatial correlations. Excluding Dynamic Mask Generation collapses the compression rate and marginally degrades accuracy, indicating that the on-demand dynamic mask not only reduces communication overhead but also filters noise to reinforce global correlation learning. Eliminating Historical Pattern Mining impairs both accuracy and compression, as its long-range cross-attention supplies critical context for dynamic embeddings and masks under heterogeneous temporal correlations.

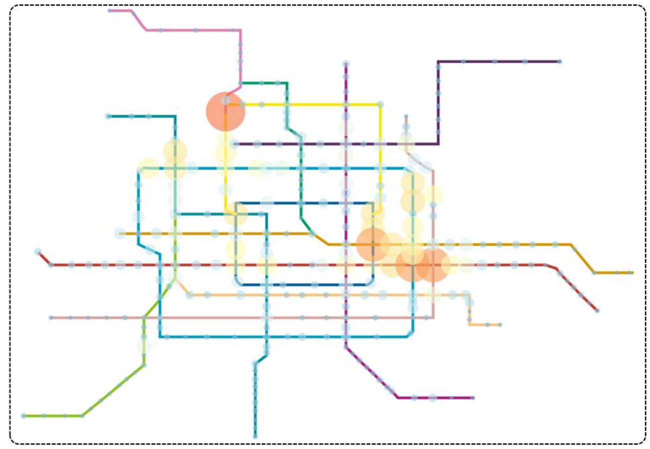
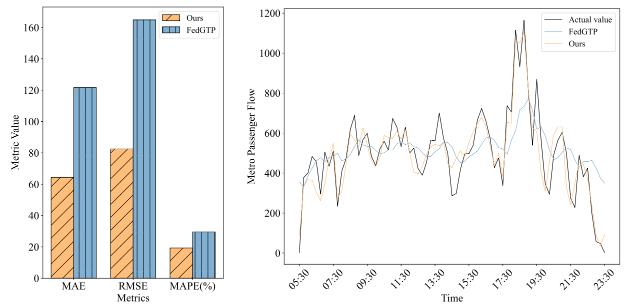


Figure 10: Visualization of Beijing metro passenger flow prediction results. Larger nodes indicate heavier congestion and red ones mark abnormal peak flow.

5 Deployment

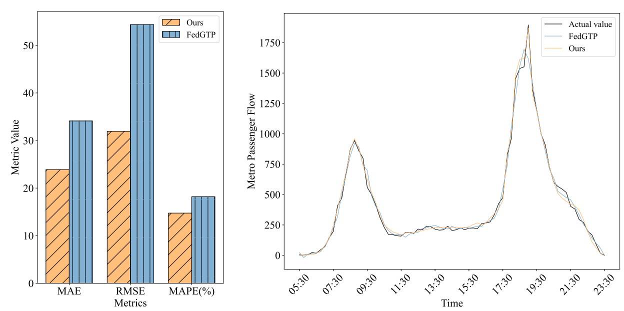
We implemented FedMetro on a socket-based architecture in which each metro line operates as an independent federated client during both training and inference. FedMetro has been successfully deployed in the Beijing metro system, providing congestion alerts that help the operator dispatch security staff and enact crowd-control measures. Additionally, FedMetro offers an interactive visualisation dashboard. As illustrated in Fig. 10, each station appears as a node whose radius and colour jointly encode its congestion level. Users can interact with the dashboard to explore detailed flow trends. To verify that station-level accuracy improves after the deployment, we comparison FedMetro with FedGTP, the current SOTA federated graph-learning method that satisfies the privacy constraints.

As shown in Fig. 11, the passenger peaks at Beijingzhan Station are aligned with train arrivals rather than the regular commuting rush hours. Consequently, its spatial correlations with neighbouring stations evolve rapidly over time. The dynamic embedding adapts to time-evolving passenger flow spatial correlations across metro stations much better than FedGTP, yielding lower prediction errors. As shown in Fig. 12, FedMetro markedly outperforms FedGTP in forecasting the evening-peak surge, demonstrating its ability to capture sudden demand anomalies promptly.



(a) Comparison of metric value **(b) Comparison of predicted and actual value**

Figure 11: Comparison of deployment performance at Beijingzhan Station.



(a) Comparison of metric value **(b) Comparison of predicted and actual value**

Figure 12: Comparison of deployment performance at Zhichunlu Station.

6 Related Work

6.1 Metro Passenger Flow Prediction

Metro passenger flow prediction is a typical spatial-temporal task in transportation systems. [28–39]. Unlike ride-hailing demand prediction [18, 20, 40], which is typically performed at the region or roadway level, it focuses on metro station nodes. Here, passenger-flow correlations are driven by line topology and transfer structures rather than by simple geometric distance. Early studies used convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture spatial-temporal patterns. However, these hybrid models are limited by Euclidean space assumptions and struggle to represent complex spatial dependencies in metro networks. The emergence of spatial-temporal graph neural networks (STGNNs) represented a significant advancement by integrating graph reasoning with temporal modeling. Early STGNNs relied on physical-topology graphs, which model the static structure of the metro network, but subsequent enhancements incorporated multi-modal data (*e.g.*, weather, air quality) [3] and predefined correlation graphs [1, 4–6]. Adaptive STGNNs went a step further by learning node-specific embeddings that allow the graph itself to emerge from data [7, 19]. Most recently, dynamic STGNNs have been proposed to track the evolving relationships among stations [8, 22, 41, 42].

However, existing metro passenger flow prediction methods predominantly adopt centralized training paradigms, which conflict with real-world privacy regulations like GDPR [9] that enforce localized data processing. This fragmentation disrupts inter-line passenger flow correlations and degrades prediction accuracy. Our proposed FedMetro system addresses this fundamental contradiction through federated graph learning, enabling global spatial correlation modeling while preserving data privacy.

6.2 Federated Graph Learning

Federated graph learning has emerged as a critical solution for privacy-preserving spatial-temporal prediction, with current research primarily focused on its application to spatial-temporal graph neural networks (STGNNs) [11]. Based on the construction paradigm of spatial correlation graphs, existing methods can be categorized into three classes: 1) Non-topology federated graph learning ignoring spatial dependencies via FedAvg-based temporal parameter aggregation [25], inherently limited in capturing global correlations [27, 43]; 2) Predefined-topology federated graph learning constructing fixed graphs through physical topologies [12–14] or community detection on public topology data [15, 16]; 3) Adaptive-topology federated graph learning implicitly modeling global spatial correlations through distributed GNN parameter aggregation and gradient-based topology propagation [17, 18, 44].

However, the aforementioned methods still require frequent transmission of high-dimensional gradient matrices during inference. In large cities like Beijing, where the metro system comprises 29 lines and 522 stations, this poses a significant communication bottleneck in practical deployment scenarios. FedMetro introduces a novel communication compression scheme using dynamic mask matrices, effectively eliminating this deployment bottleneck while maintaining spatial correlation learning capabilities.

7 Conclusion

In this paper, we propose FedMetro, a metro passenger flow prediction system based on federated graph learning, leveraging distributed Automatic Fare Collection (AFC) systems organized by metro lines. We introduce a novel federated dynamic graph learning approach to model time-evolving global spatial correlations, and employ cross-attention mechanisms to address heterogeneous temporal correlations. Additionally, we propose a dynamic mask-based communication compression method to reduce federated inference overhead. Extensive evaluations on three real-world metro AFC datasets demonstrate the effectiveness of our approach. Notably, FedMetro has been deployed in the Beijing metro system, enabling station congestion prediction and timetable optimization.

Acknowledgments

We sincerely thank the reviewers for their constructive comments. This work was partially supported by National Science Foundation of China (NSFC) (Grant Nos. U21A20516, 62425202, 62336003), the Beijing Natural Science Foundation (Z230001), the Fundamental Research Funds for the Central Universities No. JK2024-03, the Didi Collaborative Research Program and the State Key Laboratory of Complex & Critical Software Environment (SKLCCSE). Xiaoxi He, Yi Xu and Yongxin Tong are the corresponding authors.

References

[1] Pei Li, Sheng Wang, Hantao Zhao, Jia Yu, Liyang Hu, Haodong Yin, and Zhiyuan Liu. 2023. IG-Net: An Interaction Graph Network Model for Metro Passenger Flow Forecasting. *IEEE Transactions on Intelligent Transportation Systems* 24, 4 (2023), 4147–4157.

[2] Yun Bai, Qianyun Hu, Tin Kin Ho, Haiyang Guo, and Baohua Mao. 2021. Timetable Optimization for Metro Lines Connecting to Intercity Railway Stations to Minimize Passenger Waiting Time. *IEEE Transactions on Intelligent Transportation Systems* 22, 1 (2021), 79–90.

[3] Jinlei Zhang, Feng Chen, Zhiyong Cui, Yinan Guo, and Yadi Zhu. 2020. Deep learning architecture for short-term passenger flow forecasting in urban rail transit. *IEEE Transactions on Intelligent Transportation Systems* 22, 11 (2020), 7004–7014.

[4] Du Yin, Renhe Jiang, Jiewen Deng, Yongkang Li, Yi Xie, Zhongyi Wang, Yifan Zhou, Xuan Song, and Jedi S Shang. 2023. MTMGNN: Multi-time multi-graph neural network for metro passenger flow prediction. *GeoInformatica* 27, 1 (2023), 77–105.

[5] Lingbo Liu, Jingwen Chen, Hefeng Wu, Jiajie Zhen, Guanbin Li, and Liang Lin. 2020. Physical-virtual collaboration modeling for intra-and inter-station metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems* 23, 4 (2020), 3377–3391.

[6] Jingcheng Wang, Yong Zhang, Yun Wei, Yongli Hu, Xinglin Piao, and Baocai Yin. 2021. Metro passenger flow prediction via dynamic hypergraph convolution networks. *IEEE Transactions on Intelligent Transportation Systems* 22, 12 (2021), 7891–7903.

[7] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, Article 1494, 12 pages.

[8] Peng Xie, Minbo Ma, Tianrui Li, Shenggong Ji, Shengdong Du, Zeng Yu, and Junbo Zhang. 2023. Spatio-temporal dynamic graph relation learning for urban metro flow prediction. *IEEE Transactions on Knowledge and Data Engineering* 35, 10 (2023), 9973–9984.

[9] General Data Protection Regulation (GDPR). 2018. https://gdpr-info.eu.

[10] Beijing Subway. 2001. https://bjsubway.com/.

[11] Xingbo Fu, Binchi Zhang, Yushun Dong, Chen Chen, and Jundong Li. 2022. Federated Graph Machine Learning: A Survey of Concepts, Techniques, and Applications. *SIGKDD Explor. Newsl.* 24, 2 (2022), 32–47.

[12] Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. 2021. Cross-node federated graph neural network for spatio-temporal data modeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 1202–1211.

[13] Xiaoming Yuan, Jiahui Chen, Jiayu Yang, Ning Zhang, Tingting Yang, Tao Han, and Amir Taherkordi. 2022. Fedstn: Graph representation driven federated learning for edge computing enabled urban traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems* 24, 8 (2022), 8738–8748.

[14] Qingxiang Liu, Sheng Sun, Min Liu, Yuwei Wang, and Bo Gao. 2024. Online spatio-temporal correlation-based federated learning for traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems* 25, 10 (2024), 13027–13039.

[15] Tao Qi, Lingqiang Chen, Guanghui Li, Yijing Li, and Chenshu Wang. 2023. FedAGCN: A traffic flow prediction framework based on federated learning and Asynchronous Graph Convolutional Network. *Applied Soft Computing* 138 (2023), 110175.

[16] Zhongming Feng, Haoran Zha, Congan Xu, Yuanzhi He, and Yun Lin. 2024. FCGCN: Feature Correlation Graph Convolution Network for Few-Shot Individual Identification. *IEEE Transactions on Consumer Electronics* 70, 1 (2024), 2848–2860.

[17] Chenhan Zhang, Shuyu Zhang, JQ James, and Shui Yu. 2021. FASTGNN: A topological information protected federated learning approach for traffic speed forecasting. *IEEE Transactions on Industrial Informatics* 17, 12 (2021), 8464–8474.

[18] Linghua Yang, Wantong Chen, Xiaoxi He, Shuyue Wei, Yi Xu, Zimu Zhou, and Yongxin Tong. 2024. FedGTP: Exploiting Inter-Client Spatial Dependency in Federated Graph-based Traffic Prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 6105–6116.

[19] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 1907–1913.

[20] Yansheng Wang, Yongxin Tong, Zimu Zhou, Ziyao Ren, Yi Xu, Guobin Wu, and Weifeng Lv. 2022. Fed-LTD: Towards Cross-Platform Ride Hailing via Federated Learning to Dispatch. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 4079–4089.

[21] Yansheng Wang, Yongxin Tong, Dingyuan Shi, and Ke Xu. 2021. An Efficient Approach for Cross-Silo Federated Learning to Rank. In *2021 IEEE 37th International Conference on Data Engineering*. IEEE, 1128–1139.

[22] Wenying Duan, Shujun Guo, Zimu Zhou, Wei Huang, Hong Rao, and Xiaoxi He. 2025. Dynamic Localisation of Spatial-Temporal Graph Neural Network. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*. ACM, 283–294.

[23] Wenying Duan, Xiaoxi He, Zimu Zhou, Lothar Thiele, and Hong Rao. 2023. Localised Adaptive Spatial-Temporal Graph Neural Network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 448–458.

[24] Qian Lou, Yilin Shen, Hongxia Jin, and Lei Jiang. 2021. SAFENet: A Secure, Accurate and Fast Neural Network Inference. In *9th International Conference on Learning Representations*. OpenReview.net.

[25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[26] Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Vol. 33. Curran Associates Inc., 3557–3568.

[27] Lei Liu, Yuxing Tian, Chinmay Chakraborty, Jie Feng, Qingqi Pei, Li Zhen, and Keping Yu. 2023. Multilevel federated learning-based intelligent traffic flow forecasting for transportation network management. *IEEE Transactions on Network and Service Management* 20, 2 (2023), 1446–1458.

[28] Caleb Chen Cao, Yongxin Tong, Lei Chen, and H. V. Jagadish. 2013. WiseMarket: a new paradigm for managing wisdom of online social users. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*. ACM, 455–463.

[29] Chen Jason Zhang, Yongxin Tong, and Lei Chen. 2014. Where to: crowd-aided path selection. *Proc. VLDB Endow.* 7, 14 (2014), 2005–2016.

[30] Yongxin Tong, Jieying She, Bolin Ding, Libin Wang, and Lei Chen. 2016. Online mobile Micro-Task Allocation in spatial crowdsourcing. In *2016 IEEE 32nd International Conference on Data Engineering*. IEEE, 49–60.

[31] Dawei Gao, Yongxin Tong, Jieying She, Tianshu Song, Lei Chen, and Ke Xu. 2016. Top-k team recommendation in spatial crowdsourcing. In *International conference on web-age information management*. Springer, 191–204.

[32] Jieying She, Yongxin Tong, Lei Chen, and Tianshu Song. 2017. Feedback-Aware Social Event-Participant Arrangement. In *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 851–865.

[33] Qian Tao, Yuxiang Zeng, Zimu Zhou, Yongxin Tong, Lei Chen, and Ke Xu. 2018. Multi-Worker-Aware Task Planning in Real-Time Spatial Crowdsourcing. In *Database Systems for Advanced Applications*. Springer, 301–317.

[34] Tianlong Zhang, Chang Gao, Yuxiang Zeng, Shuyuan Li, Yi Xu, and Yuanyuan Zhang. 2024. Flight Planning at Scale: A Bipartite Matching Based Approach. In *Database Systems for Advanced Applications*. Springer, 19–36.

[35] Tianshu Song, Ke Xu, Jiangneng Li, Yiming Li, and Yongxin Tong. 2020. Multi-skill aware task assignment in real-time spatial crowdsourcing. *Geoinformatica* 24, 1 (2020), 153–173.

[36] Yongxin Tong, Yuxiang Zeng, Zimu Zhou, Lei Chen, and Ke Xu. 2022. Unified Route Planning for Shared Mobility: An Insertion-based Framework. *ACM Trans. Database Syst.* 47, 1, Article 2 (2022), 48 pages.

[37] Yongxin Tong, Dingyuan Shi, Yi Xu, Weifeng Lv, Zhiwei Qin, and Xiaocheng Tang. 2023. Combinatorial Optimization Meets Reinforcement Learning: Effective Taxi Order Dispatching at Large-Scale. *IEEE Transactions on Knowledge and Data Engineering* 35, 10 (2023), 9812–9823.

[38] Chang Gao, Tianlong Zhang, Yuxiang Zeng, Yi Xu, Shuyuan Li, and Yuanyuan Zhang. 2024. Swift: A Data-Driven Flight Planning System at Scale. *Proc. VLDB Endow.* 17, 12 (2024), 4465–4468.

[39] Yiming Niu, Jinliang Deng, Lulu Zhang, Zimu Zhou, and Yongxin Tong. 2025. Accurate and Efficient Multivariate Time Series Forecasting via Offline Clustering. In *2025 IEEE 41st International Conference on Data Engineering*. IEEE, 3056–3069.

[40] Shuyue Wei, Yuanyuan Zhang, Zimu Zhou, Tianlong Zhang, and Ke Xu. 2024. FedSM: A Practical Federated Shared Mobility System. *Proc. VLDB Endow.* 17 (2024), 4445–4448.

[41] Binwu Wang, Pengkun Wang, Yundong Zhang, Xu Wang, Zhengyang Zhou, Lei Bai, and Wang Yang. 2024. Towards dynamic spatial-temporal graph learning: a decoupled perspective. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 547–555.

[42] Aoyu Liu and Yaying Zhang. 2024. Spatial–Temporal Dynamic Graph Convolutional Network With Interactive Learning for Traffic Forecasting. *IEEE Transactions on Intelligent Transportation Systems* 25, 7 (2024), 7645–7660.

[43] Hanqiu Wang, Rongqing Zhang, Xiang Cheng, and Liuqing Yang. 2022. Federated Spatio-Temporal Traffic Flow Prediction Based on Graph Convolutional Network. In *14th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 221–225.

[44] Hao Miao, Yan Zhao, Chenjuan Guo, Bin Yang, Kai Zheng, and Christian S. Jensen. 2025. Spatio-Temporal Prediction on Streaming Data: A Unified Federated Continuous Learning Framework. *IEEE Transactions on Knowledge and Data Engineering* 37, 4 (2025), 2126–2140.