Parallel Interactive Attention Network for Short-Term Origin–Destination Prediction in Urban Rail Transit

Wenzhong Zhou 1,2,*, Chunhai Gao 2 and Tao Tang 1

1 School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China; ttang@bjtu.edu.cn
2 Traffic Control Technology Co., Ltd., Beijing 100070, China; chunhai.gao@bj-tct.com
* Correspondence: wenzhong_zhou@126.com

Abstract: Short-term origin–destination (termed as OD) prediction is crucial to improve the operation of urban rail transit (termed as URT). The latest research results show that deep learning can effectively improve the performance of short-term OD prediction and meet the real-time requirements. However, many advanced neural network design ideas have not been fully applied in the field of short-term OD prediction in URT. In this paper, a novel parallel interactive attention network (termed as PIANet) for short-term OD prediction in URT is proposed to further improve the short-term OD prediction accuracy. In the proposed PIANet, a novel omnidirectional attention module (termed as OAM) is proposed to improve the representational power of the network by calculating the feature weights in the channel–spatial dimension. Moreover, a simple yet effective feature interaction is proposed to improve the feature utilization. Based on the two real-world datasets from the Beijing subway, the comparative experiments demonstrate that the proposed PIANet outperforms the state-of-the-art deep learning methods for short-term OD prediction in URT, and the ablation studies demonstrate that the proposed OAMs and feature interaction play an important role in improving the short-term OD prediction accuracy.

Keywords: origin–destination prediction; urban rail transit; deep learning; attention mechanism

1. Introduction

Urban rail transit (termed as URT) plays an important role in solving the problems of traffic congestion and environmental pollution in medium and large cities. Short-term origin–destination (termed as OD) prediction is crucial to improve the operation of URT. Short-term OD prediction in URT can alleviate problems such as congestion or an incident that may occur in the URT network, so as to improve the operation of URT and provide a better travel experience for passengers.

At present, there are three main challenges for short-term OD prediction in URT: (1) data sparsity—in an OD matrix, many OD pairs are either small or zero; (2) complex spatio-temporal correlations—specifically, the variation of passenger flow among adjacent OD matrices in the time dimension usually follows the same distribution. Moreover, for each OD matrix, the change of one element may cause the change of other elements. Thus, there are complex spatio-temporal correlations among adjacent OD matrices; (3) low data availability from the same day—specifically, in-and-out card swiping information for each passenger is required to obtain a complete OD matrix. However, there are different time delays for different passengers between entering the station and leaving the station; therefore, it is difficult to obtain the historical OD matrices of the same day, and only the historical inflow series of the same day can be obtained. The above challenges make it difficult to further improve the short-term OD prediction performance.

The existing OD prediction methods in traffic scenarios can be mainly divided into three categories, which are the conventional methods [1,2], the machine learning methods [3,4], and
the recently emerging deep learning methods [5–8]. There are three main problems with the conventional methods and machine learning methods as follows: (1) the computational complexity is high for large-scale URT network, which cannot meet the need of real-time prediction; (2) it is difficult to capture complex spatio-temporal correlations; (3) the prediction accuracy is low. Compared with the conventional methods and machine learning methods, the deep learning methods can not only meet the need of real-time prediction, but also further improve the prediction accuracy.

In recent years, deep learning has developed rapidly and achieved excellent performance in various fields [9–16]. However, many advanced neural network design ideas have not been fully applied in the field of short-term OD prediction in URT. For example, channel–spatial attention and feature interaction are rarely applied in the short-term OD prediction in URT. The introduction of these advanced neural network designs into the field of short-term OD prediction in URT can further improve the short-term OD prediction accuracy by overcoming the above challenges.

Based on the above analysis, in this paper, a novel parallel interactive attention network (termed as PIANet) for short-term origin–destination prediction in URT is proposed to further improve the short-term OD prediction accuracy. The novelty of the proposed PIANet mainly lies in the architectural design, in which two innovative components including the omnidirectional attention module (termed as OAM) and the feature interaction are proposed. Therefore, our main contributions are three-fold:

- A novel omnidirectional attention module (termed as OAM) is proposed to improve the representational power of the network by calculating the feature weights in the channel–spatial dimension, which can overcome the difficulty of extracting effective feature information caused by data sparsity and complex spatio-temporal correlations.
- A simple yet effective feature interaction is proposed to improve the feature utilization between two network branches.
- Based on the two real-world datasets from the Beijing subway, the comparative experiments demonstrate that the proposed PIANet outperforms the state-of-the-art deep learning methods for short-term OD prediction in URT, and the ablation studies show that the proposed OAMs and feature interaction play an important role in improving the short-term OD prediction performance.

The rest of this paper is organized as follows. Section 2 surveys deep learning methods for short-term OD prediction in URT and the attention mechanism. Section 3 describes the proposed PIANet in detail, including problem formulation, the proposed OAM, the dense compression attention block (termed as DCAB), the overall architecture of the proposed PIANet, and the loss function. In Section 4, the basic settings including datasets, implementation details, evaluation metrics, and comparison methods are introduced. Then, the comparative experimental results among the proposed PIANet and the comparison methods are shown and analyzed. After that, the ablation studies are performed to verify the effectiveness of the proposed OAMs and feature interaction. In Section 5, this paper is summarized.

2. Related Work

2.1. Deep Learning Methods for Short-Term OD Prediction in Urban Rail Transit

As an emerging technology, deep learning has been widely used in various fields with its powerful feature extraction and expression ability. In recent years, several deep learning methods have emerged for short-term OD prediction in URT [8,17–22]. Based on the multi-source spatio-temporal data, Ref. [8] proposed a spatio-temporal long short-term memory network by improving the architecture of LSTM to improve prediction accuracy. The experimental results demonstrated that STLSTM can effectively exploit the potential information of multisource data and improve the prediction effect. Moreover, a channel-wise attentive split-convolutional neural network (termed as CASCNN) [17] was proposed to solve the data availability and data sparsity issues by using an inflow/outflow-gated mechanism and a masked loss function. Ref. [18] proposed a multi-resolution spatio-temporal neural...
network (termed as MRSTN) for real-time OD prediction, where the multi-resolution spatial feature extraction modules were first used to capture the local spatial dependencies, and ConvLSTMs were used to capture the temporal evolution of demand. The experimental results demonstrated that MRSTN outperformed the advanced deep learning methods (e.g., STResNet [6]). Ref. [19] proposed a completion augmentation-based self-attention temporal convolutional network (termed as CA-SATCN) by addressing the recent destination distribution availability, augmenting the flow presentation for each station, and digging out the global spatial dependency to improve the short-term OD prediction accuracy. Ref. [20] proposed temporal Pearson correlation coefficients, approximate entropy, and spatial correlations as indicators to reflect the inherent spatio-temporal correlations and complexity of the OD flow. Ref. [21] proposed a multi-fused residual network (termed as MF-ResNet) to capture multiple complex dependencies, in which the convolution-based residual network units model the temporal closeness, mid-term periodicity, as well as long-term periodicity features, and the fully connected layers capture external factors. Ref. [22] proposed a spatio-temporal convolutional neural network (termed as STCNN) by using OD pair importance calculation, lagged spatio-temporal relationship construction, lagged spatio-temporal learning, real-time information learning, and sequential-temporal learning blocks. The experimental results demonstrated that the STCNN outperforms the advanced methods with significant superiority on critical OD pairs.

It can be seen that there are few deep learning methods that introduce channel–spatial attention into the field of short-term OD prediction in URT. The first contribution of this paper is introducing the proposed OAM, a variant of the channel–spatial attention module, into the field of short-term OD prediction in URT. In addition, for the existing deep learning methods (e.g., CASCNN [17], CA-SATCN [19]) with several inputs, there is a problem of low feature utilization between two adjacent network branches. Thus, the second contribution of this paper is proposing a feature interaction to further improve the feature utilization.

2.2. Attention Mechanism

The attention mechanism is crucial for human perception, as it can help humans focus on the target area in the global image to obtain high-value information. In recent years, there has been plenty of research applying the attention mechanism to various fields (e.g., computer vision [23–26]). Notably, the efficient Squeeze-and-Excitation block [23] was proposed to model the cross-channel relationships by learning the channel attention. Then, a Convolutional Block Attention Module (termed as CBAM) [25] was proposed by learning channel attention and spatial attention separately. In [25], a global max pooling operation was added for the channel attention to generate a finer channel attention map, and then exploiting both spatial attention and channel attention was demonstrated to be superior to using only the channel attention. In the field of short-term OD prediction in URT, channel attention was introduced in [17] to weigh different OD inputs.

Inspired by CBAM [25], in the proposed OAM of this paper, an omnidirectional attention map is first generated by combining a channel attention map and a spatial attention map, and then used to fine-tune the importance of each element in the input feature in the channel–spatial dimension. Note that the channel–spatial dimension denotes the three-dimensional coordinate space consisting of the channel axis, height axis, and width axis.

3. Methods

In this section, the short-term OD prediction of this paper is first formulated. Then, the architectures of the OAM and DCAB are introduced in turn. After that, the overall architecture of the proposed PIANet is introduced. Lastly, the loss function is introduced.

3.1. Problem Formulation

For the short-term OD prediction of this paper, the historical OD matrices $M$ of previous days and the historical inflow series $N$ of the same day are used to generate the
predicted OD matrix $O$, where any of the historical OD matrices and any historical inflow series can be formulated as follows:

$$M^{d,t} = \begin{bmatrix}
    m^{d,t}_{11} & m^{d,t}_{12} & \cdots & m^{d,t}_{1j} & \cdots & m^{d,t}_{1\phi} \\
    m^{d,t}_{21} & m^{d,t}_{22} & \cdots & m^{d,t}_{2j} & \cdots & m^{d,t}_{2\phi} \\
    \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
    m^{d,t}_{i1} & m^{d,t}_{i2} & \cdots & m^{d,t}_{ij} & \cdots & m^{d,t}_{i\phi} \\
    \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
    m^{d,t}_{\phi1} & m^{d,t}_{\phi2} & \cdots & m^{d,t}_{\phi j} & \cdots & m^{d,t}_{\phi\phi}
\end{bmatrix}, \quad (1)$$

$$N^{t} = \begin{bmatrix}
    n^{t}_{1} \\
    n^{t}_{2} \\
    \vdots \\
    n^{t}_{i} \\
    \vdots \\
    n^{t}_{\phi}
\end{bmatrix}, \quad (2)$$

where $m^{d,t}_{ij}$ denotes the OD flow between station $i$ and station $j$ at the time interval $t$ on day $d$, and $n^{t}_{i}$ represents the inflow of station $i$ at the time interval $t$ of the same day. Given that the predicted time interval is the time interval $\kappa$ on day $\delta$, the time delay of OD data acquisition is $\alpha$ minutes, and prediction time granularity is set to $\beta$ minutes. Figure 1 shows the temporal relationship between the historical OD data and the predicted OD matrix. According to Figure 1, the short-term OD prediction can be formulated as follows:

$$O^{\kappa,\delta} = f(M^{\kappa-\delta-x_1, \delta-\alpha-\beta-x_2}, N^{\delta-\alpha-\beta-x_2}), \quad (3)$$

where $O^{\kappa,\delta}$ represents the OD matrix at the predicted time interval, and $M^{\kappa-\delta-x_1}$ denotes the historical OD matrix at the time interval $\delta - x_1$ on day $\kappa - y$, where $y = 1, 2, \cdots, \lambda_1$ and $x_1 = 0, 1, \cdots, \lambda_2 - 1$. $N^{\delta-\alpha-\beta-x_2}$ denotes the inflow series at the time interval $\delta - \alpha - \beta - x_2$ of the same day, where $x_2 = 1, 2, \cdots, \lambda_3$. $f(\cdot, \cdot)$ is a nonlinear function. In this paper, $f(\cdot, \cdot)$ denotes the forward process of the proposed PIANet.

Figure 1. Temporal relationship between the historical OD data and the predicted OD matrix.

3.2. Omnidirectional Attention Module

The OAM is proposed to fine-tune the importance of each element of the input feature in the channel–spatial dimension by generating a learnable omnidirectional attention map, so as to achieve more effective feature expression. The architecture of the proposed OAM is shown in Figure 2. As shown in Figure 2, the input feature and output feature are, respectively, $X \in \mathbb{R}^{C \times H \times W}$ and $Y \in \mathbb{R}^{C' \times H' \times W'}$, where $C$, $H$, and $W$, respectively, denote the number of channels, the height, and the width. In the proposed OAM, the omnidirectional attention map $P_1 \in \mathbb{R}^{C \times H \times W}$ is first generated by a dot-product operation between the
generated channel attention map $R_1$ and the generated spatial attention map $R_2$, and then used to fine-tune the elements in the input feature. The specific process of the proposed OAM is as follows.

**Figure 2.** The architecture of omnidirectional attention module.

Firstly, the channel attention map $R_1 \in \mathbb{R}^{C \times H \times W}$ is generated to measure the importance of each feature map in the input feature, which can be formulated as:

$$G_1 = w_1 \text{GAP}_s(X) \oplus w_2 \text{GMP}_s(X), \quad (4)$$

$$L_1 = \sigma_{lr}(g_{fc}(G_1)), \quad (5)$$

$$S_1 = \sigma_{sig}(g_{fc}(L_1)), \quad (6)$$

$$R_1 = \psi_s(S_1), \quad (7)$$

where $w_1$ and $w_2$ are the trainable weights, and $\oplus$ represents the adding operation. $\text{GAP}_s(\cdot)$ and $\text{GMP}_s(\cdot)$ denote the global average pooling operation and the global max pooling operation in the spatial dimension, respectively. Note that the spatial dimension denotes the two-dimensional coordinate space consisting of the height axis and the width axis. $g_{fc}(\cdot)$ represents the fully connected layer. $\sigma_{lr}$ and $\sigma_{sig}$ denote the leaky ReLU activation function and the sigmoid activation function, respectively. $\psi_s(\cdot)$ represents the repeat operation in the spatial dimension. In Equation (4), compared with using only one global pooling operation, using two different global pooling operations in the spatial dimension can obtain two different feature statistics, thus inferring finer channel attention. Thus, $G_1 \in \mathbb{R}^{C \times 1 \times 1}$ is generated. To reduce parameter overhead while ensuring the flexibility of learnable attention map generation, the fully connected layers in Equations (5) and (6) are used to obtain $L_1 \in \mathbb{R}^{C/r \times 1 \times 1}$ and $S_1 \in \mathbb{R}^{C \times 1 \times 1}$, respectively, where $r$ is the reduction ratio. After the repeat operation in Equation (7), the channel attention map $R_1 \in \mathbb{R}^{C \times H \times W}$ is generated.

Simultaneously, the spatial attention map $R_2 \in \mathbb{R}^{C \times H \times W}$ is generated to measure the importance of each element in the input feature in the spatial dimension, which can be formulated as:

$$G_2 = [\text{GAP}_c(X), \text{GMP}_c(X)], \quad (8)$$

$$L_2 = \sigma_{lr}(g_{bn}(g_{conv}(G_2))), \quad (9)$$

$$S_2 = \sigma_{sig}(L_2), \quad (10)$$

$$R_2 = \psi_c(S_2), \quad (11)$$

where $[\cdot, \cdot]$ denotes the concatenation operation along the channel axis, and $\text{GAP}_c(\cdot)$ and $\text{GMP}_c(\cdot)$ denote the global average pooling operation and the global max pooling operation along the channel axis, respectively. $g_{conv}(\cdot)$ and $g_{bn}(\cdot)$ denote the two-dimensional convolution and the batch normalization, respectively. $\psi_c(\cdot)$ represents the repeat operation along the channel axis. In Equation (8), compared with using only one global pooling operation, using the global average pooling operation and the global max pooling operation
along the channel axis can infer finer spatial attention. Thus, \( G_2 \in \mathbb{R}^{2 \times H \times W} \) is generated. In Equations (9) and (10), the convolutional layer and sigmoid activation function are used to obtain \( L_2 \in \mathbb{R}^{1 \times H \times W} \) and \( S_2 \in \mathbb{R}^{1 \times H \times W} \), respectively. Then, according to Equation (11), \( S_2 \) are copied along the channel axis to obtain the spatial attention map \( R_2 \in \mathbb{R}^{C \times H \times W} \). Note that the repeat operations on \( S_1 \) and \( S_2 \) are performed to facilitate the fusion between \( R_1 \) and \( R_2 \).

Eventually, the dot-product operation between \( R_1 \) and \( R_2 \) is performed to generate the omnidirectional attention map \( P_1 \), and then the dot-product operation between \( X \) and \( P_1 \) is performed to obtain \( Y \), which can be formulated as:

\[
P_1 = R_1 \odot R_2,
\]

\[
Y = X \odot P_1,
\]

where \( \odot \) denotes the dot-product operation. It can be seen that \( P_1 \) contains the feature weights of \( X \) in the channel–spatial dimension, which can fine-tune the importance of each element in \( X \) in the channel–spatial dimension by Equation (13).

### 3.3. Dense Compression Attention Block

The architecture of DCAB is shown in Figure 3. According to Figure 3, the proposed DCAB consists of a dense compression block (termed as DCB) and an OAM. In the DCAB, the proposed OAM is used to improve the feature expression by fine-tuning the importance of each element in the input feature in the channel–spatial dimension. Moreover, the DCB is composed of a dense block [11] and a convolutional layer. In the DCB, the dense block is used to improve the training efficiency. It can be seen that the number of channels in the output feature of the dense block is drastically increased compared to the number of channels in its input features. Therefore, in order to reduce the computational complexity, a convolutional layer is subsequently used to compress the dimension in the channel axis.

![Figure 3. The architecture of the dense compression attention block.](image)

### 3.4. Parallel Interactive Attention Network

The proposed PIANet takes the historical OD matrices \( M \in \mathbb{R}^{\lambda_1 \times \phi \times \phi} \) of previous days and the historical inflow series \( N \in \mathbb{R}^{\lambda_1 \times \phi} \) of the same day as inputs and the predicted OD matrix \( O \in \mathbb{R}^{1 \times \phi \times \phi} \) as output, where \( \lambda = \lambda_1 \lambda_2 = \lambda_3 \) is set. Figure 4 shows the overall architecture of the proposed PIANet, and Table 1 presents the PyTorch-like pseudocode for the forward process of the proposed PIANet. According to Figure 4 and Table 1, the proposed PIANet can be divided into three stages: stage 1 is the generation of the historical OD matrices of the same day, stage 2 is the feature interaction, and stage 3 is the generation of the predicted OD matrix.

![Figure 4. The overall architecture of the parallel interactive attention network.](image)
Table 1. PyTorch-like pseudocode for the forward process of the proposed PIANet.

<table>
<thead>
<tr>
<th>Stage 1: Generation of The Historical OD Matrices of The Same Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $D_1 = \text{DCAB} (M_1)$; $D_2 = \text{DCAB} (D_1)$;</td>
</tr>
<tr>
<td>2: $B_1, B_2 = \text{unsqueeze} \ (-1)$. repeat (1, 1, $\phi$);</td>
</tr>
<tr>
<td>3: $M_1 = B_1 \odot \text{torch.softmax} (D_2 + \text{DCAB} (B_1))$;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 2: Feature Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>4: $D_3 = \text{DCAB} (D_1)$;</td>
</tr>
<tr>
<td>5: $E_1 = \text{torch.concat} (D_4, D_5)$; $A_2 = D_4 + D_5$;</td>
</tr>
<tr>
<td>6: $D_6 = \text{DCAB} (E_1)$; $D_7 = \text{DCAB} (A_2)$;</td>
</tr>
<tr>
<td>7: $A_3 = D_6 + D_7$; $E_2 = \text{torch.concat} (D_8, D_9)$;</td>
</tr>
<tr>
<td>8: $D_4 = \text{DCAB} (A_2)$; $D_9 = \text{DCAB} (E_2)$;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 3: Generation of The Predicted OD Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>9: $E_3 = \text{torch.concat} (D_4, D_9)$;</td>
</tr>
<tr>
<td>10: $O = \text{DCB} (E_3)$;</td>
</tr>
</tbody>
</table>

Output: The predicted OD matrix $O$.

3.4.1. Generation of the Historical OD Matrices of the Same Day

As shown in Figure 4, stage 1 can be described in two steps. The first step is to generate the OD distribution $S_1$ of the historical OD matrices of the same day by introducing the OD distribution information contained in the historical OD matrices of the previous days. The second step is to generate the historical OD matrices $M_1$ of the same day by using $S_1$. The specific process is as follows.

Firstly, $M$ is fed into two DCABs in turn to obtain the feature $D_1 \in \mathbb{R}^{\lambda \times \phi \times \phi}$ and $D_2 \in \mathbb{R}^{\lambda \times \phi \times \phi}$, respectively, where $D_2$ can provide effective OD distribution information for the generation of the historical OD matrices of the same day.

Simultaneously, $N$ is fed into the unsqueeze layer and the repeat layer in turn to obtain $U_1 \in \mathbb{R}^{\lambda \times \phi \times 1}$ and $B_1 \in \mathbb{R}^{\lambda \times \phi \times \phi}$, respectively. It can be seen that the dimension of $B_1$ is the same as that of $M$. However, $B_1$ is obtained by copying the inflow series in $U_1$ along the width axis; therefore, $B_1$ does not contain any OD flow information. After that, $B_1$ is fed into a DCAB for feature extraction to obtain $D_3 \in \mathbb{R}^{\lambda \times \phi \times \phi}$. In order to extract effective OD flow feature information, the adding operation between $D_2$ and $D_3$ is performed to obtain $A_1 \in \mathbb{R}^{\lambda \times \phi \times \phi}$. Then, $A_1$ is fed into the softmax layer to obtain $S_1 \in \mathbb{R}^{\lambda \times \phi \times \phi}$, where the softmax layer can perform normalization computation along the width axis. Thus, $S_1$ can be regarded as the OD distribution of the historical OD matrices of the same day, where row $i$ of each $\phi \times \phi$ matrix in $S_1$ represents the OD distribution from station $i$ to any station.

Secondly, the dot-product operation between $B_1$ and $S_1$ is performed to generate the historical OD matrices $M_1 \in \mathbb{R}^{\lambda \times \phi \times \phi}$ of the same day, which can be formulated as:

$$M_1 = B_1 \odot S_1.$$  \hspace{1cm} (14)

3.4.2. Feature Interaction

In stage 2, the feature information in $D_1$ and $M_1$ is further extracted separately, and then the extracted feature information is used interactively to improve the feature utilization, which is helpful to improve the short-term OD prediction accuracy. The specific process is as follows.

$D_1$ and $M_1$ are fed into the DCABs for feature extraction to obtain $D_4 \in \mathbb{R}^{\lambda \times \phi \times \phi}$ and $D_5 \in \mathbb{R}^{\lambda \times \phi \times \phi}$, respectively, where $D_4$ and $D_5$ contain the feature information of the historical OD matrices of previous days and the historical OD matrices of the same day, respectively. In order to improve the feature utilization and provide richer feature expression, two different feature fusion operations are performed on $D_4$ and $D_5$ to obtain $E_1 \in \mathbb{R}^{2\lambda \times \phi \times \phi}$ and $A_2 \in \mathbb{R}^{\lambda \times \phi \times \phi}$, respectively, which can be expressed as:

$$E_1 = [D_4, D_5],$$  \hspace{1cm} (15)
$$A_2 = D_4 \oplus D_5.$$  \hspace{1cm} (16)

In order to further improve the feature expression, the feature interaction is performed again. Specifically, $E_1$ and $A_2$ are fed into the DCABs to generate $D_6 \in \mathbb{R}^{\lambda \times \phi \times \phi}$ and
\( D_7 \in \mathbb{R}^{\lambda \times \phi \times \phi} \), respectively. Then, the adding operation and concatenation operation are performed on \( D_6 \) and \( D_7 \) to obtain \( A_3 \in \mathbb{R}^{\lambda \times \phi \times \phi} \) and \( E_2 \in \mathbb{R}^{2\lambda \times \phi \times \phi} \), respectively, which can be formulated as:

\[
A_3 = D_6 \oplus D_7, \quad \text{(17)}
\]

\[
E_2 = [D_6, D_7]. \quad \text{(18)}
\]

After that, \( A_3 \) and \( E_2 \) are fed into the DCABs for deeper feature extraction to obtain \( D_8 \in \mathbb{R}^{\lambda \times \phi \times \phi} \) and \( D_9 \in \mathbb{R}^{\lambda \times \phi \times \phi} \), respectively.

### 3.4.3. Generation of the Predicted OD Matrix

After obtaining \( D_8 \) and \( D_9 \), the concatenation operation between \( D_8 \) and \( D_9 \) is performed to obtain \( E_3 \in \mathbb{R}^{2\lambda \times \phi \times \phi} \). Eventually, \( E_3 \) is fed into a DCB to generate the predicted OD matrix \( \hat{O} \).

### 3.5. Loss Function

In this paper, \( L_2 \) loss is used as the loss function, which can be formulated as

\[
L_2 = \frac{1}{\phi^2} \sum_{i=1}^{\phi} \sum_{j=1}^{\phi} (\hat{M}_{ij} - \hat{O}_{ij})^2, \quad \text{(19)}
\]

where \( \hat{O} \) is the predicted OD matrix obtained by removing the dimension whose size is 1 from \( O \); \( \hat{M}_{ij} \) and \( \hat{O}_{ij} \) denote the value of row \( i \) and column \( j \) in the real OD matrix and the predicted OD matrix, respectively.

### 4. Experiments and Results

In this section, the basic settings including datasets, implementation details, evaluation metrics and comparison methods are first introduced. Then, the experimental results are presented and discussed, in which the experiments include two parts: (1) comparison of short-term OD prediction performance among the proposed PIANet and the three comparison methods; and (2) ablation studies.

#### 4.1. Basic Settings

##### 4.1.1. Datasets

Two real-world large-scale OD datasets from the Beijing subway were used in the experiments, which are presented in Table 2. According to Table 2, these two OD datasets were collected in 2021 and 2022, named BJSubway2021 and BJSubway2022, respectively. For BJSubway2021, the date range is from 2021.9.14 to 2021.10.17 and from 2021.12.14 to 2021.12.29, the number of data records is 137 million, the number of stations is 369, and thus the OD matrix dimension is \( 369 \times 369 \). BJSubway2021 contains 50 days, of which 36 days, 7 days, and 7 days were used as training set, validation set, and test set, respectively. For BJSubway2022, the date range is from 2022.1.1 to 2021.1.9 and from 2022.1.18 to 2022.2.21, the number of data records is 98 million, the number of stations is 372, and thus the OD matrix dimension is \( 369 \times 369 \). BJSubway2022 contains 44 days, of which 36 days, 6 days, and 7 days were used as training set, validation set, and test set, respectively. For both BJSubway2021 and BJSubway2022, the time range in a day was set from 6:00 to 23:00; each record contains the entry station ID, the exit station ID, the entry time, and the number of passengers entering the station; And the time interval is 5 min.
Table 2. Data description.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BJSubway2021</th>
<th>BJSubway2022</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>14 September 2021 to 17 October 2021</td>
<td>1 January 2022 to 9 January 2022</td>
</tr>
<tr>
<td></td>
<td>14 December 2021 to 29 December 2021</td>
<td>18 January 2022 to 21 February 2022</td>
</tr>
<tr>
<td>Data record</td>
<td>137 million</td>
<td>98 million</td>
</tr>
<tr>
<td>Station Number</td>
<td>369</td>
<td>372</td>
</tr>
<tr>
<td>Matrix Dimension</td>
<td>369 × 369</td>
<td>372 × 372</td>
</tr>
<tr>
<td>Time</td>
<td>6:00 to 23:00</td>
<td>6:00 to 23:00</td>
</tr>
<tr>
<td>Time Interval</td>
<td>5 min</td>
<td>5 min</td>
</tr>
</tbody>
</table>

4.1.2. Implementation Details

In this paper, the proposed PIANet was implemented by using PyTorch and Python on an NVIDIA GeForce RTX 3090 with 24GB memory, and the learning rate and batch size were set to 0.001 and 16, respectively. The hyperparameters were set as follows: both \( w_1 \) and \( w_2 \) were initialized to 0.5, and \( a = 15, \beta = 15, \lambda_1 = 3, \lambda_2 = 3, \lambda_3 = 9, r = 2 \). Moreover, the Adam optimizer was used to optimize the parameters in the proposed PIANet. The early stopping was used as the training stopping criterion to avoid overfitting, that is, the best validation RMSE was recorded in each epoch, and the training did not stop until the best validation RMSE rose 10 times in a row. All convolutional layers were followed by the batch normalization layers and leaky ReLu layers. Furthermore, the historical OD matrices of previous days and the historical inflow series of the same day were preprocessed by Min-Max normalization to avoid gradient explosion before feeding into the architecture of the proposed PIANet.

4.1.3. Evaluation Metrics

In this paper, the root mean square error (termed as RMSE), mean absolute error (termed as MAE), and symmetric mean absolute percentage error (termed as SMAPE) were used as evaluation metrics to evaluate the short-term OD prediction performance, in which RMSE, MAE, and SMAPE are, respectively, formulated as

\[
RMSE = \sqrt{\frac{1}{\phi^2} \sum_{i=1}^{\phi} \sum_{j=1}^{\phi} (\hat{M}_{ij} - g(\hat{O}_{ij}))^2},
\]

\[
MAE = \frac{1}{\phi^2} \sum_{i=1}^{\phi} \sum_{j=1}^{\phi} |\hat{M}_{ij} - g(\hat{O}_{ij})|,
\]

\[
SMAPE = \frac{1}{\phi^2} \sum_{i=1}^{\phi} \sum_{j=1}^{\phi} \frac{|\hat{M}_{ij} - g(\hat{O}_{ij})|}{(\hat{M}_{ij} + g(\hat{O}_{ij}))/2 + c},
\]

where the notations \( \hat{M}_{ij} \) and \( \hat{O}_{ij} \) are the same as those in Equation (19), \(|·|\) denotes \( L_1 \) norm, and \( c \) is a very small positive constant to prevent the denominator from being 0, which is set as \( 10^{-9} \). Since \( \hat{O}_{ij} \) usually contains decimals and negative numbers, in order to evaluate the model performance more accurately, \( \hat{O}_{ij} \) was evaluated after rounding and non-negative integer processing. Thus, \( g(·) = g_2(g_1(·)) \) is a composite function, in which \( g_1(·) \) denotes the rounding function and \( g_2(·) \) denotes the function of taking nonnegative integers. \( g_2(g_1(\hat{O}_{ij})) \) can be expressed as:

\[
g_2(g_1(\hat{O}_{ij})) = \begin{cases} 
0, & g_1(\hat{O}_{ij}) < 0 \\
g_1(\hat{O}_{ij}), & g_1(\hat{O}_{ij}) \geq 0.
\end{cases}
\]

According to Equations (20) and (21), RMSE and MAE are two different evaluation metrics that can measure the absolute error between the predicted OD matrix and the real OD matrix but cannot reflect whether the predicted OD matrix is underestimated or
overestimated relative to the real OD matrix. According to Equation (22), SMAPE as a relative error metric can reflect whether the predicted OD matrix is underestimated or overestimated relative to the real OD matrix. Therefore, the prediction performance of the network can be comprehensively evaluated by using RMSE, MAE, and SMAPE.

4.1.4. Comparison Methods

In this paper, three state-of-the-art deep learning methods for short-term OD prediction in URT including ConvLSTM [27], STResNet [6], and CASCNN [17] were selected as the comparison methods of the proposed PIANet. Compared with the original paper, the implementation details of the three comparison methods were fine-tuned to make them more suitable for the dataset in this paper. Specifically, in terms of use of input data, the inputs of ConvLSTM and STResNet are the same as that of the proposed PIANet. For CASCNN, the inflow and outflow series of the same day and the finished OD matrices of previous days were used the same as the original paper. The batch sizes of ConvLSTM, STResNet, and CASCNN were 16, 32, and 32, respectively. Moreover, for ConvLSTM, STResNet, and CASCNN, the learning rates and loss functions were set to 0.001 and $L_2$ loss, respectively.

4.2. Results and Discussions

In this section, the proposed PIANet is firstly compared with the comparison methods to show the superiority of the proposed PIANet in short-term OD prediction performance. Then, ablation studies are conducted to investigate the impacts of the proposed OAMs and feature interaction on the prediction performance of the proposed PIANet.

4.2.1. Comparison of Prediction Performance

In order to show the excellent short-term OD prediction performance of the proposed PIANet, comprehensive comparisons among the proposed PIANet and the three comparison methods were conducted. Table 3 shows the comparisons of RMSE, MAE, and SMAPE among the proposed PIANet and the three comparison methods (i.e., ConvLSTM, STResNet, and CASCNN) in BJSubway2021 and BJSubway2022. Note that all values of RMSE, MAE, and SMAPE are displayed in the form of mean ± std (standard deviation) in Table 3 and Table 4.

Table 3. Comparisons of RMSE, MAE, and SMAPE among the proposed PIANet and the three comparison methods in BJSubway2021 and BJSubway2022.

<table>
<thead>
<tr>
<th>Method</th>
<th>BJSubway2021</th>
<th>BJSubway2022</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>MAE</td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>0.648 ± 0.343</td>
<td>0.161 ± 0.100</td>
</tr>
<tr>
<td>STResNet</td>
<td>0.722 ± 0.476</td>
<td>0.172 ± 0.119</td>
</tr>
<tr>
<td>CASCNN</td>
<td>0.740 ± 0.473</td>
<td>0.178 ± 0.116</td>
</tr>
<tr>
<td>PIANet</td>
<td>0.524 ± 0.191</td>
<td>0.145 ± 0.084</td>
</tr>
</tbody>
</table>

* Bold for the evaluation metric represents the best result.

As shown in Table 3, the proposed PIANet can obtain lower RMSE, lower MAE, and lower SMAPE than the comparison methods in BJSubway2021 and BJSubway2022, which indicates that the proposed PIANet performs better than the comparison methods in short-term OD prediction performance. Specifically, compared with the suboptimal results, the RMSE, MAE, and SMAPE obtained by the proposed PIANet are, respectively, reduced by 19.1%, 9.9%, and 4.3% in BJSubway2021, and the RMSE, MAE, and SMAPE obtained by the proposed PIANet are, respectively, reduced by 20.3%, 14.7%, and 6.6% in BJSubway2022.

In order to present more detailed comparison results, the comparisons of RMSE, MAE, and SMAPE at each time interval during the morning peak hours (i.e., 7:30–9:10) among the proposed PIANet and the comparison methods (i.e., ConvLSTM, STResNet, and CASCNN) in BJSubway2021 and BJSubway2022 are shown in Figure 5 and Figure 6, respectively.
As shown in Figure 5a, the RMSE obtained by ConvLSTM is lower than those obtained by STResNet and CASCNN at each time interval in BJSubway2021. According to Figure 6b,c, the MAE and SMAPE obtained by ConvLSTM are, respectively, lower than those obtained by STResNet and CASCNN at each time interval in BJSubway2022. However, according to Figure 5b,c, the MAE and SMAPE obtained by ConvLSTM do not maintain the performance advantage compared with STResNet and CASCNN at each time interval in BJSubway2021. Specifically, in Figure 5b, the MAE obtained by ConvLSTM is lower than those obtained by STResNet and CASCNN before 8:30, and then higher than those obtained by STResNet and CASCNN after 8:30. In Figure 5c, the SMAPE obtained by ConvLSTM is lower than those obtained by STResNet and CASCNN before 8:15, and then higher than those obtained by STResNet and CASCNN after 8:15. Similarly, as shown in Figure 6a, the RMSE obtained by ConvLSTM does not maintain the performance advantage compared with STResNet and CASCNN at each time interval in BJSubway2022.

The above results show that none among ConvLSTM, STResNet, and CASCNN can always maintain the advantages in short-term OD prediction performance under the three evaluation metrics (i.e., RMSE, MAE, and SMAPE). However, it can be seen from Figures 5 and 6 that the RMSE, MAE, and SMAPE obtained by the proposed PIANet are lower than those obtained by the comparison methods at all time intervals in BJSubway2021 and BJSubway2022, which further verifies the superiority of the proposed PIANet in short-term OD prediction performance.

In addition, the comparisons of convergence curves of best validation RMSE among the proposed PIANet and the comparison methods in BJSubway2021 and BJSubway2022 are shown in Figure 7. According to Figure 7a, CASCNN and STResNet stop converging to the lower minima at the 7th epoch and the 15th epoch, respectively. The convergence of ConvLSTM is relatively smooth; however, the convergence speed is slow. Compared with the three comparison methods, the proposed PIANet can converge faster and can stably converge to a lower best RMSE. According to Figure 7b, the proposed PIANet and the three comparison methods can all converge smoothly. Notably, compared with the
three comparison methods, the proposed PIANet can still converge faster and still stably converge to a lower best RMSE.

![Image](image_url)

**Figure 7.** Comparisons of convergence curves of best validation RMSE among the proposed PIANet and the comparison methods in BJSubway2021 and BJSubway2022.

The possible reasons for the better short-term OD prediction performance of the proposed PIANet compared to ConvLSTM, STResNet, and CASCNN are as follows: In terms of feature extraction, ConvLSTM can capture the temporal correlation well, but it cannot capture the spatial correlation. STResNet only uses a stack of convolutional layers for short-term OD prediction, and thus it is difficult to extract effective non-local feature information. The architecture of CASCNN is relatively simple; therefore, it is difficult to extract effective feature information for complex spatio-temporal correlations. In the proposed PIANet, the proposed OAMs are used to improve the representational power of the network by calculating the feature weights in the channel–spatial dimension, which can extract effective feature information for complex spatio-temporal correlations. In terms of feature utilization, none of the comparison methods have feature interactions between two network branches, which makes the feature utilization between two network branches low. However, the feature interaction between two network branches is used in the proposed PIANet, which further improves feature utilization between two network branches.

In short, compared with the comparison methods, the proposed PIANet can stably converge to a lower best validation RMSE during training; thus, the proposed PIANet shows better short-time OD prediction performance during testing.

### 4.2.2. Ablation Studies

The OAMs and feature interaction are proposed to further improve the short-term OD prediction performance. To verify the effectiveness of the proposed OAMs and feature interaction, four extra experiments were conducted.

In order to verify the effectiveness of the proposed OAMs, three experiments were conducted to form three variants of the proposed PIANet: (1) PIANet-A—all DCABs in Figure 4 are changed to DCBs; (2) PIANet-B—only DCABs in stage 2 of Figure 4 are changed to DCBs; (3) PIANet-C—only DCABs in stage 1 of Figure 4 are changed to DCBs. Moreover, in order to verify the effectiveness of feature interaction, the feature interaction in stage 2 of Figure 4 is removed to form a variant of the proposed PIANet, named PIANet-D.

The comparisons of RMSE, MAE, and SMAPE among the proposed PIANet and its variants (i.e., PIANet-A, PIANet-B, PIANet-C, and PIANet-D) in BJSubway2021 and BJSubway2022 are presented in Table 4. Furthermore, the comparisons of convergence curves of best validation RMSE among the proposed PIANet and its variants in BJSubway2021 and BJSubway2022 are shown in Figure 8.
Table 4. Comparisons of RMSE, MAE, and SMAPE among the proposed PIANet and its variants in BJSubway2021 and BJSubway2022.

<table>
<thead>
<tr>
<th>Method</th>
<th>BJSubway2021</th>
<th>BJSubway2022</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>MAE</td>
</tr>
<tr>
<td>PIANet-A</td>
<td>0.558 ± 0.230</td>
<td>0.153 ± 0.088</td>
</tr>
<tr>
<td>PIANet-B</td>
<td>0.556 ± 0.234</td>
<td>0.151 ± 0.089</td>
</tr>
<tr>
<td>PIANet-C</td>
<td>0.550 ± 0.238</td>
<td>0.153 ± 0.096</td>
</tr>
<tr>
<td>PIANet-D</td>
<td>0.610 ± 0.292</td>
<td>0.158 ± 0.096</td>
</tr>
<tr>
<td>PIANet</td>
<td>0.524 ± 0.191 *</td>
<td>0.145 ± 0.084</td>
</tr>
</tbody>
</table>

* Bold for the evaluation metric represents the best result.

According to Table 4, the proposed PIANet shows better short-term OD prediction performance with lower RMSE, lower MAE, and lower SMAPE than PIANet-B and PIANet-C, and then PIANet-B and PIANet-C perform better than PIANet-A in general, except that the SMAPE of PIANet-C is higher than that of PIANet-A in BJSubway2021. The above comparison results show that both the OAMs in stage 1 and the OAMs in stage 2 play an important role in improving the short-term OD prediction performance. And the more OAMs, the better the short-term OD prediction performance. Furthermore, the proposed PIANet outperforms PIANet-D, which indicates that feature interaction plays an important role in improving the short-term OD prediction performance.

According to Figure 8, the proposed PIANet can converge to a lower best validation RMSE than PIANet-B and PIANet-C in BJSubway2021 and BJSubway2022, and PIANet-B and PIANet-C can converge to a lower best validation RMSE than PIANet-A in BJSubway2021 and BJSubway2022. Moreover, the proposed PIANet can converge to a lower best validation RMSE than PIANet-D in BJSubway2021 and BJSubway2022. The comparison results of convergence curves in Figure 8 further verify the analytical results of Table 4.

In a word, it can be seen from Table 4 and Figure 8 that the proposed OAMs and feature interaction are effective in improving the short-term OD prediction performance.

5. Conclusions

In this paper, the proposed PIANet is introduced to further improve the short-term OD prediction accuracy in URT. The proposed PIANet involves two innovative components, namely, the OAMs and feature interaction. The OAMs are proposed to improve the representational power of the network by calculating the feature weights in the channel–spatial dimension, and the feature interaction is proposed to improve the feature utilization. Based on the two real-world datasets from the Beijing subway, according to the comparative experiments, the proposed PIANet obtains lower RMSE, lower MAE, and lower SMAPE than the comparison methods, which indicates that the proposed PIANet outperforms the comparison methods in short-term OD prediction performance. According to the ablation
studies, the proposed OAMs and feature interaction play an important role in improving the short-term OD prediction performance.

The embedding of the proposed OAMs and the feature interaction makes the proposed PIANet have excellent feature extraction ability. Therefore, the proposed PIANet is not only suitable for the URT system of Beijing, but also can be extended to the URT system in other cities. Specifically, when the PIANet is extended to another URT system with different characteristics and scale, only the sizes of input and output need to be modified according to the scale of the new URT system, and then the corresponding input data and output data can be used for training. Due to its excellent feature extraction ability, the PIANet can capture the new passenger flow distribution during training. Therefore, after training, the PIANet can adapt to the new URT system and achieve accurate short-term OD prediction.

Author Contributions: Conceptualization, W.Z., C.G., and T.T.; methodology, W.Z.; software, W.Z.; validation, W.Z.; formal analysis, W.Z.; investigation, W.Z.; resources, W.Z. and C.G.; data curation, C.G.; writing—original draft preparation, W.Z.; writing—review and editing, W.Z.; visualization, W.Z.; supervision, C.G. and T.T.; project administration, C.G. and T.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the security of passenger flow information.

Conflicts of Interest: Author Wenzhong Zhou was employed by Traffic Control Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.