ODformer: Spatial-Temporal Transformers for Long Sequence Origin-Destination Matrix Forecasting Against Cross Application Scenario

Jin Huang\(^a\), Bosong Huang\(^b\), Weihao Yu\(^b\), Jing Xiao\(^b\), Ruzhong Xie\(^b\), Ke Ruan\(^b\)\(^,*\)

\(^a\)South China Normal University, Guangzhou, China
\(^b\)Research Institute of China Telecom Corporate Ltd., Guangzhou, China

Abstract

Origin-Destination (OD) matrices record directional flow data between pairs of OD regions. The intricate spatiotemporal dependency in the matrices makes the OD matrix forecasting (ODMF) problem not only intractable but also non-trivial. However, most of the related methods are designed for very short sequence time series forecasting in specific application scenarios, which cannot meet the requirements of the variation in scenarios and forecasting length of practical applications. To address these issues, we propose a Transformer-like model named ODformer, with two salient characteristics: (i) the novel OD Attention mechanism, which captures special spatial dependencies between OD pairs of the same origin (destination), greatly improves the ability of the model to predict cross application scenarios after combining with 2D-GCN that captures spatial dependencies between OD regions. (ii) a PeriodSparse Self-attention that effectively forecasts long sequence OD matrix series while adapting to the periodic differences in different scenarios. Generous experiments in three application backgrounds (i.e., transportation traffic, IP backbone network traffic, crowd flow) show our method outperforms the state-of-the-art methods.

Keywords: Origin-destination matrix, Graph convolutional network, Sparse self-attention mechanism, Long sequence time-series forecasting

1. Introduction

Origin-Destination (OD) matrices are widely used to provide location-based services (LBSs) or to aid in data mining and discovery in various scenarios related to geographic information. Specifically, OD matrices help to estimate travel expense \(^\[1\]\) and schedule trips, so Google Maps provide OD matrices for better developing LBSs. From a broader perspective, OD matrices support various data analysis tasks by introducing geographic information. For instance, the COVID-19 spread in a country could be forecasted according to the human movements OD matrix \(^\[2\]\). Moreover, internet traffic OD matrices show the potential to support IP backbone network management \(^\[3\]\) and internet resource allocation \(^\[4\]\). In a word, the specific meaning of data stream is diverse between different application backgrounds, but we can still unify them into one cross-scenario application. Generally speaking, OD matrices record every traffic flow from origin regions to destination regions which are geographically divided. The regions of OD matrices could be city spans, country spans, or even entire continents spans. Each element \((i, j)\) in OD matrices is an OD pair that represents a certain traffic flow from region \(i\) to region \(j\) against a diverse background, as illustrated in Figure \(^\[1\]\).

Here, we consider the Origin-Destination matrix forecasting problem (ODMF) in which the elements represent traffic data for different application backgrounds (e.g., transportation traffic, IP backbone network traffic). There are many application scenarios for ODMF results. For example, in the field of transportation, it can be used to plan the itinerary in advance, the allocation of traffic resources, and in the field of IP network, it can be used to help efficiently complete tasks such as network planning or operation and maintenance.

Existing methods are designed against exceedingly fixed application backgrounds and under limited problem settings. On the one hand, those methods are specific to a single application \(^\[5\] \[6\] \), thus, all the algorithm details intend to achieve the best performance on a single application scenario which leads to poor portability. Further, they are invalid for revealing the intrinsic spatial-temporal features in OD matrix-type data across different application scenarios. On the other hand, many methods \(^\[7\] \[8\] \[9\] design the experiment setting just to fulfill literally excellent results but ignore the actual demands of ODMF, such as long sequence time-series forecasting (LSTF). As the sampling interval is around 5 to 20 minutes in most OD matrix datasets, existing methods only predict the next several or even one timestep, which means predicting the next several minute’s OD matrices. For resource allocation or another downstream task, this kind of prediction goes for nothing in a realistic application. Therefore, we focus on the long sequence ODMF problem, which could be defined as follows: in multiple scenarios, based on OD matrices at several historical timesteps, we predict OD matrices at longer future timesteps, illustrated in Figure \(^\[2\]\).
To address the above problems, we present a spatial-temporal transformer (ODformer) for long sequence OD matrices forecasting. To the best of our knowledge, it is the first time to introduce the transformer to the ODMF problem, meanwhile with targeted improvements to accommodate the problem. We propose a more general spatial dependency module that consists of the OD Attention mechanism and the 2D graph neural network, which captures spatial dependencies in multiple scenarios. Further, we redesign the self-attention mechanism of the transformer named as PeriodSparse Self-attention to reduce the computational complexity of the long-sequence prediction tasks, while adapting to time series with diverse features from different scenarios. The contribution of this paper are summarized as follows:

- ODformer is the first Transformer-like model for the ODMF problem. Meanwhile, for the first time, it crosses multiple scenarios in ODMF, covering three real-world application scenarios: transportation traffic, crowd flow, and IP backbone network traffic.
- We propose OD Attention mechanism to mine the specialized spatial dependency between OD pairs of the same origin (destination), which is dramatically different from existing ODMF methods (i.e., merely considering the topological correlation between OD regions). Based on this, we achieve significant improvement in capturing spatial-temporal dependency in various application scenarios.
- To support long sequence forecasting, we propose PeriodSparse Self-attention mechanism to capture long-range temporal correlation by achieving sparse attention with \(O(L \ln L)\) time complexity and adapting to periodicity in different scenarios.

2. Related works

2.1. OD Matrices Forecasting in Specific Applications

We consider ODTF problems under different scenarios in turn: transportation traffic [8, 1, 10, 11, 12, 6, 5], crowd flow [13, 2], IP backbone network traffic [14, 15, 16, 7, 9]. Below, we will analyze the sophisticated differences between diverse application backgrounds in the ODTF problem.

In the transportation traffic forecasting scenario, the data reveal the highest relationships with the geographic factors. Thus the point is how to capture spatial dependency. The direct idea is to transfer the 2D convolution to the spatial domain, and extract the spatial dependency in a way similar to processing pictures. For instance, STGCN [8] formulates the traffic spatial-temporal prediction task on graph structures. MDL [17] designs two sophisticated models for predicting node flow and edge flow, respectively, based on fully convolutional networks to integrate transportation geographic factors. Despite convolutional neural networks (CNNs) could capture simple adjacent dependency between regions, the depth topological relationship between geographical regions is different from that in Euclidean space extracted by CNN. With the proposal of Graph Neural Network (GCN) [18], some graph-based models [8, 5], capture the topological information to forecast vehicle traffic flow between regions by employing GCNs that work in a non-Euclidean space. In summary, the existing methods in the background of transportation traffic mainly focus on capturing spatial dependency by extracting the topological adjacent information.
In the crowd flow forecasting scenario, the available data reveals different features of transportation background. The crowd data [13] is usually collected from smart devices carried by humans that provide GPS information. In this case, the crowd data is more fine-grained and shows less relevance to road traffic structure. Due to the more complex spatial dependency, merely depending on geographic adjacency is far from enough. OD-CRN [2] proposes the Dynamic Graph Constructor to capture highly temporal-dependent relationships between OD regions.

In the IP backbone network traffic forecasting scenario, the spatial dependency varies from the aforementioned. As the topology structures of IP networks consist of artificially set network nodes, the spatial dependency is comparatively irrelevant to natural geographic factors. Most existing methods [14] [16] [7] treat the network traffic volumes between IP nodes as grid-like data, thus applying convolution neural networks to extract spatial patterns. MSTNN [13] simply transfers the graph-based method to the IP network data to try to capture the spatial dependency, just like in geography determined scenario, which ignores the particularity of artificial IP network nodes.

2.2. Long Sequence Time-series Forecasting

In the time-series forecasting problem, the ability to capture precise long-term dependency [19] [20] between historical sequence and predicted sequence determines the availability degree against the real-world application. Long sequence time-series forecasting (LSTF) advocates a longer predicted sequence than historical [21].

The traditional sequence-to-sequence LSTF models mostly apply recurrent neural networks (RNNs) [22] such as LSTM [23] or GRU [24] to capture temporal patterns. The multilayered LSTM [23] maps the variable length input sequence to a vector of fixed dimensionality, then decodes the relatively long target sequence from the vector. And the RNN Encoder–Decoder [21] proposes a gated recursive convolutional neural network to avoid the model degrading extremely rapidly when dealing with relatively long sequences. But the above RNN methods perform ineffectiveness when the prediction sequence becomes longer, which is probably due to the intrinsic limitations of recurrent neural network architectures [25].

Inducement of attention mechanism to time series forecasting problem brings a chance to process long sequence. Dual-Stage Attention-Based Recurrent Neural Network [25] [19] adaptively extracts relevant input features at each time step by referring to the previous encoder hidden state, then selects relevant encoder hidden states across all time steps. With the amelioration of canonical Transformer architecture [26] against locality-agnostics and memory bottleneck [27], addressing the LSTF problem by transformers became possible. Autoformer [20] adopts the sparse versions of point-wise self-attention for long series efficiency, meanwhile settling down the information utilization bottleneck. Informer [28] designed a ProbSparse Self-attention mechanism, which achieves $O(L\log L)$ in time complexity and memory usage.

3. Problem Definition

At first, we define several data forms that are crucial in the long sequence ODMF problem while trying our best to preserve generality.

As we consider the ODMF problem against diverse application scenarios, the raw data varies in forms which mostly consist of two types: traffic trajectories and traffic matrices.

**Definition 1.** (Origin-Destination Pair and Origin-Destination Matrix): Typically, a trajectory (e.g. taxi trajectory, bus trajectory, human trajectory) is a time sequence while each timeslot contains a longitude-latitude coordinate that is denoted as $[(t_1, l_1), (t_2, l_2), \ldots, (t_n, l_n)]$, where each location $l$ is represented by a longitude-latitude coordinate and $t$ is the timestep. To obtain OD pairs, we segment each trajectory as follows:

$$ [(t_1, l_1), (t_2, l_2), \ldots, (t_n, l_n)] \xrightarrow{\text{segment}} \{[(0_1, d_1), t_1], \ldots, [(o_n, d_n), t_n]\},$$

where the trajectory is segmented to $n$ OP pairs. $o_d$ represent the origin and destination regions. For improving generality, we take the departure timeslot of each trajectory segment as the $t_o$ of the OD pairs, the approximation mainly due to (i) mostly, the trajectory segment time interval is less than the interval between OD matrices’ timeslot. (ii) some trajectory data are the trip forms [5], which means the trajectory contains only one departure time and one arrival time.

In some cases, the raw data is in the form of a traffic matrix (i.e. a set of OD pairs [7] [6]), which is even easier. After the division of the area into regions (i.e., districts in cities, provinces in countries, IP internet nodes), we could assign each OD pair to the elements in the OD matrix at timeslot $T$:

$$ M_{t}^{ij} = \{(o, d, t) | o_{\text{location}} \in \mathbb{D} \cap d_{\text{location}} \in \mathbb{D} \cap t = T \} $$

The OD matrix is defined as $M_{t}^{ij} \in \mathbb{R}^{n \times N \times F}$, where the first two dimensions range over the origin and destination regions (i.e., $\mathbb{D}, \mathbb{D}$). The last dimension is denoted as the feature of each OP pair that in most cases is a single scalar, but for the sake of particular cases (e.g. multiple features [5]), we reserve that dimension.

**Definition 2.** (long sequence ODMF): After the above process, we have the input (i.e. the historical sequence at length $l$):

$$ \{M_{t-l}, \ldots, M_{t}\} $$

The output is the predicted sequence at length $O$: $\{M_{t+1}, M_{t+2}, \ldots, M_{t+O}\}$.

The definitions of "long sequence" differ in different long sequence time-series forecasting methods. The "long sequence" in this paper has two meanings: (i) $O$ is significantly longer than $I$ as primary LSTF methods [23] [28] [20] encourage. (ii) $I$ is relatively longer than most settings in other methods [2] [3] [7].
4. Methodology

4.1. Overall Architecture

As shown in Figure 3, we capture the spatial dependency before applying the attention mechanism in the temporal dimension. In each spatial dependency module, we apply the 2D-GCN and OD Attention mechanism separately to capture spatial dependency upon different aspect (i.e., the adjacent relationships between regions and intrinsic dependency among the OD pairs which shares the same origin or destination). The output of the spatial dependency module is the aggregated results of the above two by the spatial share coefficient $\alpha$, which varies according to different datasets. To complete sparse sampling for the two sparse attention mechanisms (i.e., OD Attention and PeriodSparse Self-attention), Periodicity Extractor dynamically compute the period sets with $k$ periods based on the output of the spatial dependency module. In this way, the output of the encoder contains both the spatial and temporal dependency of the OD matrix sequence, and finally, we complete the cross attention calculation together with the decoder.

4.2. Spatial Dependency

The spatial dependencies in different scenarios are quite different. For example, in transportation traffic, the spatial dependency is strongly correlated with geographic information, while in the IP network, the spatial dependencies are more manifested in the relationship between OD pairs. Based on this, we propose a spatial dependency module that combines two spatial dependencies mechanisms (i.e, OD Attention and PeriodSparse Self-attention). Periodicity Extractor dynamically compute the period sets with $k$ periods based on the output of the spatial dependency module. In this way, the output of the encoder contains both the spatial and temporal dependency of the OD matrix sequence, and finally, we complete the cross attention calculation together with the decoder.

two spatial dependencies are dynamically adjusted by the spatial share coefficient $\alpha$ according to different application scenarios.

4.2.1. Origin-Destination (OD) Attention Mechanism:

Previous works [5, 2, 6, 15] capture the spatial correlation between origin regions or destination regions, but they roughly take all the OD pairs from the same origin or destination as the same while ignoring the dependency between OD pairs from the same regions. Inspired by the spatial attention mechanism [29], we propose the OD Attention mechanism to capture the dependencies between OD pairs of the same origin (destination). At timestep $T$, the OD matrix: $M_F \in \mathbb{R}^{N \times N \times F}$ could be denoted as the origin vector: $X_F = [X_1, X_2, \ldots, X_N]^T$, $X_n \in \mathbb{R}^{N \times F}$ or destination vector $Y_F = [Y_1, Y_2, \ldots, Y_N], Y_n \in \mathbb{R}^{N \times F}$. To avoid the quadratic time complexity on the region’s number of the canonical self-attention [30], which significantly restrains the capability in LSTF, here we propose a spatial sparse self-attention mechanism.

(1) Sparse Self-attention: The brief strategy is to select the dominant query vector that contributes the most information. We apply the Shannon Entropy [31] to select the one with the smallest value (e.g. the Shannon Entropy of uniform distribution is the highest, and the more unlike the query vector and uniform distribution, the better). Since the uniform distribution is to assign the even importance to each element, which is meaningless). Take the origin attention as an example, the $i$-th query’s attention to all the keys is defined in a probability form: $p(k_i | q_i) = e^{q_i(k_i) / \sum_{j=1}^{N} e^{q_i(k_j)}},$ where $\hat{q}_i(k, k)$ is the exponential kernel: $\exp\left(\frac{q_i k}{\sigma^2}\right)$, thus the importance score on
between origin region $i$ and element $\Omega_i$. To normalize the origin attention coefficients within each region, the element $\Omega_{i,i'}$ in the attention matrix reveals the relevance level between origin region $i$ to another origin region $i'$. The (3) Destination Attention: In the almost same process, we symmetrically compute the destination attention coefficient on top of $\mathcal{Y}_{\mathcal{T}}$ and $\mathcal{Y}_i$: \[
abla_{i,i'}' = \sigma\left((\mathcal{Y}_{\mathcal{T}}W_1)W_2(\mathcal{W}_3\mathcal{Y}_i)^T + b_3\right) \tag{7}
abla_{i,i'} = \frac{\exp(\nabla_{i,i'}')}{\sum_{i'=1}^{N} \exp(\nabla_{i,i'}')} \tag{8}
\]
At last, we apply the origin attention and destination attention matrix to the OD matrix: $\mathbf{M}_{ij}^{\mathcal{T}} = \mathcal{Y}_{\mathcal{T}}\mathbf{M}_{ij}\mathcal{Y}_i$. Therefore, each element in the OD matrices is dynamically given the impact of the elements from the same origin or destination regions as illustrated in Figure 4(b).

4.2.2. 2D Graph Convolution Network:
Besides the intrinsic dependency between OD pairs, the topological relationship among diverse OD regions still plays a large or small role according to the different application scenarios. In ODMF problems, graph convolution neural networks (GCNs) have wildly used to study the connectivity and globality based on topology graphs of regions.

Here, we adopt MGCNN to capture the above relationships, which is an extended two-dimensional transformation of the GCN with Chebyshev polynomial approximation.

$$
\mathbf{H}^{(l+1)} = \sigma\left(\sum_{i=0}^{K-1} \sum_{j=0}^{K-1} T_L(\mathbf{L}_{\text{origin}}) \times \mathbf{H}^{(l)} \times T_j(\mathbf{L}_{\text{destination}}) \times W_{ij}\right) \tag{9}
$$
\[\mathbf{L}_{\text{origin}} \text{ or } \mathbf{L}_{\text{destination}} \text{ is the Laplacian matrix obtained from } \mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}, \mathbf{D} \text{ is the degree matrix, } \mathbf{I} \text{ is the identity matrix. } T_i(\mathbf{L}) \text{ is the Chebyshev polynomial of order } i. \]
is the hidden state of layer $l$ while $\mathbf{H}^{(0)}$ is the OD matrix. After the 2D-GCN operation, we get the OD matrices: $\mathbf{M}_{FF}^l$ in which the neighbor information propagates $K$ hops between origin regions or destination regions.

4.3. Temporal Dependency

Long sequence forecasting of OD matrices mainly faces two challenges: (i) the intricate temporal pattern across the OD matrices. (ii) computation efficiency, which is even more matter due to the data scale of the OD matrices. Thus, we propose the Period Sparse Self-attention, which targets the above issue that captures intrinsic temporal dependency and reduces computation consumption at the same time. However, the periodicity in the differences scenario dataset differs a lot (e.g. the sampling interval is 15mins in the GÉANT dataset while 1 day in the Japan Human Trajectory dataset). For generality against different scenarios and to capture deep temporal dependency, we propose the Periodicity Extractor to adaptively give the period set: $\mathcal{P} = \{P_1, \cdots, P_k\}$ of the inputs sequence.

4.3.1. Periodicity Extractor:

For the purpose of adaptively extracting the period length of different datasets properly, inspired by STL [34], we filter the tendency component. Given an OD matrix sequence that has gone through the spatial dependency module: $S = \{\mathbf{M}_{(T-s+1)}, \cdots, \mathbf{M}_{(T-1)}, \mathbf{M}_{T}\}$, the periodicity component is computed as follows.

$$S_{\text{trend}} = \text{MovAvr}_{\text{win}}(S)$$
$$S_{\text{period}} = S - S_{\text{trend}}$$

we apply centered moving average (window size is MovAvr$_{\text{win}}$) to get the tendency component, the remainder component is omitted for simplicity. For each sequence of length $L$: $S_T(T$ is the last timeslot), we adopt the Auto-Correlation Mechanism [20] to evaluate the chance of periodicity being $P$ based on it’s periodicity component: $\hat{S}_T$.

$$\mathcal{R}_{\hat{S}_T}(P) = \lim_{L \to \infty} \frac{1}{L} \sum_{t=1}^{L} \hat{S}_T \hat{S}_{T-P}$$
$$\mathcal{P} = \{P_1, \cdots, P_k\} = \text{Topk}_{P\in\{1, \cdots, L\}}(\mathcal{R}_{\hat{S}_T}(P))$$

where $\mathcal{R}_{\hat{S}_T}(P)$ is the time-delay similarity between $\hat{S}_T$ and its $P$ lag series $\hat{S}_{T-P}$ that could be calculated by Fast Fourier Transforms (FFT) [35]. Topk() meant to get the periods of top $k$ time-delay similarity.

4.3.2. PeriodSparse Self-attention:

Unlike the canonical self-attention mechanism (Figure 5a) with $O(L^2)$ time complexity or previous sparse attention that neglect different data time interval (Figure 5b), we propose the PeriodSparse Self-attention mechanism (Figure 5c) to adaptively choose the sparse scheme according to different data. Here we define the connectivity pattern $I = \{I_1, \cdots, I_k\}$, where $I_i$ is the set of the input vector indices which attends the $i$th output vector. Unlike the full self-attention that allows every element to attend to all positions, the PeriodSparse Self-attention is determined by the subsets of $I_i$: $A^{(k)}_i \subset \{j : i \leq j \leq i\}$. Based on the extracted period sets $\mathcal{P} = \{P_1, \cdots, P_k\}$:

$$A_i^{(k)} = \{j : (i-j) \text{ mod } P_k = 0\}$$

Notice that we set the number of attention heads according to $k$ (i.e., the indices of periods) so that we capture the spatial dependency at different periodicity by different attention heads. For every single head, we compute the output vector at each position:

$$a(S_i, A_i) = \text{softmax} \left( \frac{(W_qS_i)K_i^j}{\sqrt{d}} \right) V_{A_i}$$

where $K_i = (W_iS_{\mathcal{P}})_{j\in\mathcal{P}}, V_{A_i} = (W_iS_{\mathcal{P}})_{j\in\mathcal{P}}$ and $W_q, W_k, W_v$ are to transform an input $S$ to query, key, or value while $d$ is the inner dimension of the queries and keys. We set $k = c \cdot \max(k', \ln L)$, which makes the layer memory usage of PeriodSparse Self-attention maintains $O(L \ln L)$, $c$ is the sparsity control factor.

5. Experiments

5.1. Data

We perform experiments on five datasets in three application scenarios (i.e., IP backbone network traffic, crowd flow, transportation traffic) to evaluate the predictive ability of the proposed model across scenarios.

IP backbone network traffic: In this scenario, we apply two datasets: the GÉANT [1] dataset and the Abilene [2] dataset. The former is collected from the Abilene network, including 11 nodes (regions) and 48,097 time intervals that differ by five minutes, and the latter is collected from the pan-European research and development.

---

1. https://network.geant.org/
2. https://math.bu.edu/people/kolaczyk/datasets.html
education network GÉANT, including 23 routing nodes (regions) and 10,772 time intervals that differ by 15 minutes. The input of data includes OD matrix series and network topology graphs between nodes (regions).  

**Crowd flow:** We adopt the Japan Human Trajectory (JHT) dataset to evaluate the models in the crowd flow scenario. The OD regions are the 47 prefectures of Japan, the related topological map is the adjacency map of Japanese provinces, and the time sampling interval is one day (2020/1/1 ~ 2021/2/28).

**Transportation traffic:** To evaluate the models in traffic scenarios, we adopt two taxi trajectory datasets: New York City (NYC) dataset, which records taxi trips collected from 2013-11-01 to 2013-12-31 from Manhattan (67 subregions), the time interval is 15 minutes. Chengdu (CD) dataset: records from 1.4 billion taxi trips collected from 2014-08-03 to 2014-08-30 in Chengdu (79 regions), China.

![image](https://example.com/image.png)

Table 1: Long sequence ODMF results on five datasets (three scenarios)

| Methods  | Metric | GÉANT | Abilene | JHT | NYC | CD |
|----------|--------|-------|---------|-----|-----|----|
|          |        | 96    | 192     | 288 | 384 |    |
| LSTM     | MAE    | 2.653 | 2.928   | 3.527 | 1.321 | 1.431 | 1.593 |
|          | MSE    | 1.538 | 1.742   | 1.993 | 0.825 | 0.923 | 1.062 |
| ARIMA    | MAE    | 2.033 | 2.972   | 3.039 | 1.175 | 1.285 | 1.330 |
|          | MSE    | 1.239 | 1.487   | 1.623 | 0.623 | 0.678 | 0.685 |
| N-BEAT   | MAE    | 2.677 | 2.980   | 3.203 | 1.205 | 1.219 | 1.279 |
|          | MSE    | 1.502 | 1.536   | 1.854 | 0.637 | 0.649 | 0.672 |
| DeepAR   | MAE    | 1.924 | 2.072   | 2.133 | 1.924 | 2.072 | 2.133 |
|          | MSE    | 0.962 | 1.003   | 1.138 | 0.591 | 0.619 | 0.637 |
| STGCN    | MAE    | 2.498 | 2.832   | 3.317 | 1.289 | 1.299 | 1.324 |
|          | MSE    | 1.490 | 1.528   | 1.734 | 0.679 | 0.690 | 0.698 |
| GEML     | MAE    | 2.302 | 2.538   | 2.791 | 1.148 | 1.203 | 1.239 |
|          | MSE    | 1.173 | 1.297   | 1.418 | 0.647 | 0.682 | 0.691 |
| ODCRN    | MAE    | 2.587 | 2.781   | 2.970 | 1.238 | 1.246 | 1.301 |
|          | MSE    | 1.395 | 1.452   | 1.672 | 0.630 | 0.652 | 0.662 |
| MPGCN    | MAE    | 1.741 | 2.088   | 2.365 | 1.132 | 1.168 | 1.230 |
|          | MSE    | 0.933 | 1.085   | 1.259 | 0.631 | 0.655 | 0.669 |
| Informer | MAE    | 4.309 | 4.540   | 4.583 | 1.173 | 1.202 | 1.269 |
|          | MSE    | 1.518 | 1.693   | 1.793 | 0.611 | 0.618 | 0.632 |
| Autoformer | MAE | 1.855 | 1.986   | 2.015 | 0.971 | 0.992 | 1.039 |
|          | MSE    | 0.945 | 0.989   | 0.992 | 0.602 | 0.610 | 0.627 |
| ODformer | MAE    | 1.623 | 1.735   | 1.843 | 0.936 | 0.961 | 0.997 |
|          | MSE    | 0.908 | 0.937   | 0.940 | 0.578 | 0.583 | 0.602 |

5.2. Experimental Details

5.2.1. Baselines:

We implement two classes of baselines to compare and evaluate our proposed model on the long sequence ODMF task:  

**From the perspective of the ODMF problem:** (1) STGCN [8]: ST-GCN is one of the earliest models that induce graph convolution network (GCN) to the spatiotemporal transportation prediction problems. (2) GEML [11]: One of the state-of-the-art ODMF models in transportation background. (3) MPGCN [6]: The first model that applied 2D-GCN with multiple graphs to the ODMF problem. (4) ODCRN [2]: The state-of-the-art ODMF model in crowd flow background, with Dynamic Graph Constructor (DGC) module.

**From the perspective of the LSTM problem:** (5) LSTM [36]: One of the most common recurrent neural networks. (6) ARIMA [37]: Classic benchmark model that captures seasonality adequately. (7) N-BEAT [38]: (8) DeepAR [39]: Standard time series methods. (9) Informer [28]: One of the most advanced long sequence prediction models based on ProbSparse Self-attention mechanism. (10) Autoformer [20]: An advanced long sequence prediction model with Auto-Correlation mechanism.

3https://github.com/deepkashiwa20/ODCRN/tree/main/data  
4https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page  
5https://goo.gl/3VsEyn
Table 2: Comparison of PeriodSparse Self-attention and full self-attention.

| Input length I | 48 | 96 | 192 | 384 |
|----------------|----|----|-----|-----|
| prediction length O | 96 | 192 | 384 | 768 | 1536 | 3072 |

| Method       | MSE       | MAE       |
|--------------|-----------|-----------|
| Full Attention | 0.978 1.194 1.397 0.999 1.078 - 1.152 - - - - -  |
|             | 0.539 0.611 0.793 0.623 0.650 - 0.690 - - - - -  |
| ProbSparse Attention | 0.947 0.981 1.184 0.936 0.997 1.359 0.983 1.320 1.086 1.238 1.492 |
|               | 0.413 0.489 0.560 0.578 0.602 0.673 0.632 0.687 0.779 0.951 1.083 1.170 |

1. We replace the PeriodSparse Self-attention in ODformer with full self-attention.
2. The "-" indicates the out-of-memory.

Table 3: Ablation of OD Attention or 2D-GCN module in ODformer.

| Method       | Metric | GÉANT(48) | Abilene(96) | JHT(7) | YNC(12) | CD(12) |
|--------------|--------|-----------|-------------|--------|---------|--------|
| ODformer     | MSE    | 1.623 1.735 0.936 0.961 0.163 0.186 0.815 0.901 0.704 0.788 |
|              | MAE    | 0.908 0.937 0.578 0.583 0.277 0.310 0.484 0.512 0.209 0.210 |
| ODformer†    | MSE    | 1.690 1.867 0.955 0.971 0.169 0.198 0.819 0.909 0.710 0.793 |
|              | MAE    | 0.926 0.951 0.589 0.599 0.286 0.317 0.490 0.520 0.212 0.217 |
| ODformer§    | MSE    | 1.630 1.742 0.942 0.966 0.167 0.190 0.834 0.913 0.711 0.799 |
|              | MAE    | 0.912 0.941 0.585 0.590 0.282 0.311 0.497 0.527 0.214 0.223 |

1. ODformer† removes OD Attention from ODformer.
2. ODformer§ removes 2D-GCN from ODformer.

5.2.2. Setting:

We use a threshold of 98% to clip overly large data to remove outliers. Then, fill in the missing values based on topological neighbors of the OD regions: $M_{u,v} \leftarrow \text{AGGREGATE}([M_{u,v} | \forall u \in N(u)] \cup [M_{u,v} | \forall v \in N(v)])$, where AGGREGATE is the aggregator function [40] that aggregates information from one-hop neighbors, $N(v)$ denoted the immediate neighborhood of vertex $v$. To normalize the data, we take natural logarithm of the elements in the original OD matrices. The training/validation/test dataset split ratio is: 6:2:2. The initial learning rate is set to $10^{-4}$, using the ADAM [41] optimizer. The batch size is set to 16. The training process is early stopped within 8 epochs. All experiments are repeated five times, implemented in PyTorch [42], and conducted on a single NVIDIA GeForce RTX 3090 GPU. At last, we evaluate the overall performance of models on the long sequence ODMF by two metrics: MSE (Mean Square Error), MAE (Mean Absolute Error).

5.3. Results and Analysis

5.3.1. Overall performance:

Table 1 compares the overall performance between the baselines and proposed ODformer on the long sequence ODMF task in data of three scenarios (The input length is on the right side of the dataset). We gradually prolong the prediction length $O$ to verify the long sequence prediction ability of the model. The length setting depends on the specific scale of OD matrices so that long sequence ODMF is tractable on one single GPU for every method. Optimal and suboptimal results are shown in red and purple, respectively. For Table 1, we summarize as follows: (1) In the setting with the smallest prediction length, ODformer greatly improves the inference performance for vast majority scenarios, which illustrates the success of capturing spatial dependencies across scenarios. (2) The prediction error of ODformer rises smoothly and slowly over the growing prediction length $O$, further outperforming the rest of the models as the $O$ grows. This demonstrates ODformer’s success in improving the predictive power towards long sequence OD matrices. (3) The sub-optimal models in different scenarios are different, but our models can all perform optimally, which shows that ODformer successfully solves the problem of the ODMF cross scenario. (4) On the first set of Chengdu data, the performance of ODformer is slightly weaker than that of GEML, which may due to the spatial relationships in this dataset all depending on the adjacency information between regions.

5.3.2. Ablation study:

To demonstrate the effectiveness of our innovative modules, we performed ablation experiments as follows. (1) Removal of PeriodSparse Self-attention: The experiment was performed on the Abilene dataset. In Table 2, PeriodSparse Self-attention outperforms the full attention mechanism under various $O-I$ settings. At the same time, it also significantly outperforms full attention in memory consumption.
5.3.3. Parameter sensitivity:

We perform parameter sensitivity analysis on the spatial dependency and temporal dependency modules of the ODformer model, as shown in Figure 6.

(1) Spatial share coefficient: We conduct prediction experiments with the shortest I–O setting on each of the five datasets to analyze the spatial share coefficient $\alpha$. To better represent the trend, we use Min-Max Normalization to process the raw MSE values. We find that the $\alpha$ at the lowest MSE is different on different datasets. On more geographically related datasets, the smaller $\alpha$, the smaller the MSE. Therefore, in the end, we choose the appropriate $\alpha$ according to different application scenarios. When $\alpha$ approaches extreme values, MSE will rise sharply, which shows the necessity of introducing two different spatial dependencies.

(2) Sparsity control factor: We set the input sequence length to 192 on the Abilene dataset, and the sparsity control factors were taken from $\{0.25, 0.5, 1, 2, 4, 8\}$ for experiments. It can be observed that when the control factor is too large, the overall MSE is lower, which proves the necessity of introducing sparse attention for long sequence prediction. In practical applications, we set the control factor to 2.

5.3.4. Efficiency analysis

We analyze the time and memory complexity of ODformer here. The complexity in time and memory of ODformer for time-series length $L$ is $O(L \log L)$, and the complexity in time and memory of OD Attention on each timestep is $O(N \log N)$ (N is the number of regions). For fairness of comparison, we combine the temporal sparse attention models with full spatial attention mechanisms. Figure 7 shows the superiority of ODformer in terms of time and memory consumption.

6. Conclusion

In this paper, we study the long sequence OD matrix forecasting problem and propose ODformer to forecast long OD matrix series. Specifically, we design the OD Attention mechanism and PeriodSparse Self-attention to deal with the challenges of the variability of spatial-temporal dependencies in various application scenarios and LSTF of OD matrices. Experiments on real-world data in diverse scenarios demonstrate the effectiveness of ODformer in improving the capability of long sequence OD matrix forecasting problems.

7. Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grants 62177015.

References

[1] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, Y. Liu, Multi-task representation learning for travel time estimation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1695–1704.
