Spatio-Temporal Traffic Prediction Using Crossover Attention for Communications and Networking

Ke He, Thang X. Vu, Symeon Chatzinotas, and Björn Ottersten

The Interdisciplinary Centre for Security, Reliability and Trust (SnT) - University of Luxembourg,
L-1855 Luxembourg. E-mail: {ke.he, thang.vu, symeon.chatzinotas, bjorn.ottersten}@uni.lu

***Abstract*—This paper investigates the spatio-temporal multivariate time series prediction problem, which has important applications in various real-world tasks including network traffic modeling, network slicing, and channel estimation. To tackle this problem, many attention-based models have been proposed in the literature to predict the output of future time slots. However, we notice that the majority of attention models was designed to capture input dependency structures in only a single domain (typically the temporal domain), which limits their prediction accuracy. To solve this issue and further improve the performance of attention-based models, we propose a novel crossover attention mechanism in this paper. The crossover attention can be understood as a learnable regression kernel which prioritizes the input sequence with both spatial and temporal similarities and extracts relevant information for generating the output of future time slots. Simulation results based on realistic datasets show that when replacing the vanilla attention with the proposed crossover attention, considerable improvement on the prediction accuracy can be achieved for the existing attention-based models.**

***Index Terms*—Spatio-temporal, multivariate time series, traffic prediction, crossover attention, transformer model, deep learning.**

I. Introduction

Spatio-temporal multivariate time series (ST-MTS) refers to sequences that combine multiple variables with both spatial and temporal dimensions. ST-MTS prediction aims to predict future values across multiple related variables by combining the dimensions of time and space. Unlike univariate time series prediction, which focuses solely on temporal patterns, this approach considers both temporal dependencies and spatial interactions. This technique nowadays finds important applications in mobile traffic analysis [1] and network slicing [2], [3], multiple-input multiple-output (MIMO) channel estimation [4], [5], intelligent transportation [6], and urban planning [7], where understanding both temporal and spatial dynamics is crucial for the accurate prediction. For instance, accurate and timely mobile traffic prediction is crucial for intelligent network slicing [8], leading to reduced network congestion and enhanced service quality. When applying to MIMO channel prediction and antenna selection, it can help mitigate the channel aging issue [4], reduce the pilot overhead [9] and thereby improve the system performance.

Improving the prediction accuracy of ST-MTS generally relies on the pattern recognition of spatio-temporal dependencies. It is not surprising that traditional statistical methods, such as historical average (HA) and auto regressive integrated moving average (ARIMA), perform poorly in this task as they were designed solely for capturing the temporal correlation of time series. Nevertheless, thanks to the tremendous success of deep learning techniques during the past five years, many learning driven approaches have been investigated to tackle this issue. Such attempts originally started with the utilization of convolutional neural networks (CNNs) [10] and recurrent neural networks (RNNs) [11], which were adopted to handle spatial correlations and temporal correlations respectively.

While CNNs work well with Euclidean data structures, i.e., images or well-ordered sensors grids, they are typically not good at non-Euclidean data structures where the order of nodes and edges can vary, such as satellite networks and road networks. To efficiently exploiting the spatial dependency of non-Euclidean MTS, graph convolutional networks (GCNs) based prediction architectures have recently attracted much research interests. For instance, researchers in [12] formulated the moving terminals as a time-evolving graph, and adopted GCN to predict the future mobile traffic data. The experiments showed that their proposed GCN-based model outperforms the CNN based models in different prediction metrics. In [13], the authors also employed GCN-based model to achieve better traffic flow prediction performance than CNN-based models.

Analyzing and utilizing temporal patterns are also vital for ST-MTS prediction tasks. However, it has been shown in the literature that conventional RNN structures such as long-short term memory (LSTM) are not efficient when dealing with long-range dependencies [14]. To tackle this issue, attention-based Transformers have been recently proposed [3]. One of the key benefits of Transformers lies in their capability to capture long-range dependencies and interactions [15]. This feature comes from the fundamental attention mechanism inside Transformers, and is particularly appealing for time series modeling, resulting in significant advancements across various time series applications. For instance, the authors in [16] proposed spatio-temporal Transformer networks (STTN) to predict the traffic flows, in which the experiment results showed significant prediction accuracy improvement compared with convolutional LSTM (ConvLSTM) [17]. The authors in [2] employed Transformer structures for spatio-temporal cellular traffic prediction, and achieved a better prediction performance than RNN-based models. The encoder-decoder Transformer has also shown advantages in MIMO channel

prediction [18].

With the ever evolving development of attention models in ST-MTS prediction tasks, we notice that the state-of-the-art prediction models generally combine the attention mechanism with GCNs [15], [19]. This common practice leverages the innate abilities of attention layers and GCN layers to capture temporal and spatial correlations separately, whereas we argue that the ability of capturing structured spatial information was however ignored in practice for the existing attention mechanisms. This is because they were designed based on the principle of querying values by temporal similarities [19], which somehow limits their capability.

Motivated by these considerations, we hereby propose a novel crossover attention mechanism for attention-based models, which not only captures the temporal dynamics but also exploits the spatial dynamics. It functions as a learnable regression kernel that predicts values by simultaneously considering both spatial and temporal similarities. This feature is particularly appealing for spatio-temporal predictive learning, as it prioritizes input sequences with both spatial and temporal similarities, extracting relevant information for generating future outputs. Our simulation results on various real-world datasets, *Milan* (mobile traffic prediction) [20] and *SanDiego* (traffic flow prediction) [6], show that when simply replacing the vanilla attention module with the crossover attention, the existing attention-based models can achieve considerable improvement on the prediction accuracy in various ST-MTS prediction tasks.

II. Problem Formulation and Attention Mechanism

A. Multivariate Time Series prediction

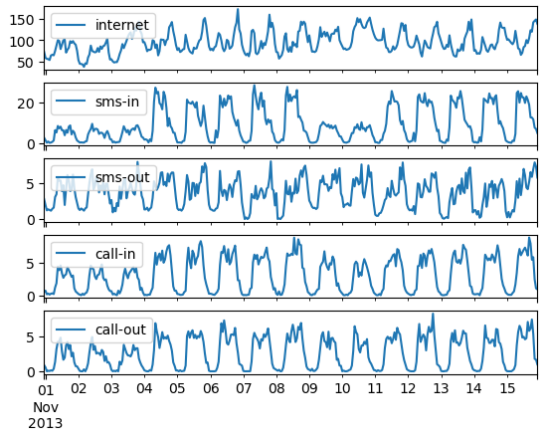
Let $\mathbf{Z}_{1:T} = \{\mathbf{z}_t\}_{t=1}^{T} \in \mathbb{R}^{T \times D_z}$ be a multivariate time series, where $\boldsymbol{z}_t = [z_{t,1}, \ldots, z_{t,i}, \ldots, z_{t,D_z}] \in \mathbb{R}^{D_z}$ is the vector of variables observed at time t, and $\mathbf{Z}_{t_1:t_2} \in \mathbb{R}^{(t_2-t_1+1)\times D_z}$ the all values at the time slice $t \in [t_1, t_2]$. The time series $\mathbf{Z}_{1:T}$ is associated with a covariate (independent) sequence denoted by $\mathbf{X}_{1:T} = \{\mathbf{x}_t\}_{t=1}^{T} \in \mathbb{R}^{T \times D_x}$ with the same notation manner, in which each vector $\mathbf{x}_t \in \mathbb{R}^{D_x}$ can contain both time-varying or static domain-specified features. The objective of MTS prediction is to predict the conditional distribution

$$\mathbf{Z}_{t+1:t+h} \sim p(\mathbf{Z}_{t+1:t+h} | \mathbf{Z}_{1:t}, \mathbf{X}_{1:t+h}; \boldsymbol{\theta}), \tag{1}$$

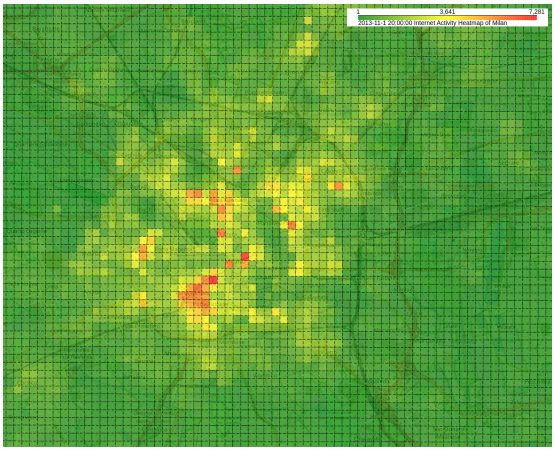
where θ denotes the parameters of a (probabilistic) prediction model. The prediction model is then used to forecast the future multi-step (h steps) multivariate targets $\mathbf{Z}_{t+1:t+h}$ based on the historical target observations $\mathbf{Z}_{1:t}$ of the past t steps as well as the associated covariates $\mathbf{X}_{1:t+h}$ until the prediction horizon. One may also want to learn a mapping of input features to the prediction model's parameters as

$$\boldsymbol{\theta} = \Psi\left(\mathbf{Z}_{1:t}, \mathbf{X}_{1:t+h}; \boldsymbol{\omega}\right), \tag{2}$$

where $\Psi\left(\cdot; \boldsymbol{\omega}\right)$ is usually a neural network parameterized by a set of learning parameters $\boldsymbol{\omega}$, e.g., weights and bias. Note that $\Psi\left(\cdot; \boldsymbol{\omega}\right)$ is often used to learn the dependency structure among the time series.



(a) Hourly cellular traffic pattern in two weeks of the $(50, 50)$-th cell in the city.



(b) Heatmap of the internet activities of all cells.

Fig. 1: Spatio-temporal cellular mobile traffic in Milan, Italy.

B. Spatio-Temporal Dynamics

In many real world MTS prediction tasks, the time series often presents spatial and temporal correlations. In general, the spatio-temporal dependencies of MTS should be utilized to improve the prediction accuracy. To give a concrete example and elaborate this, Fig. 1 is provided to demonstrate the spatio-temporal effects of cellular traffic data in city Milan. The cellular traffic data come from the public dataset released by Italia Telecom [20]. The city is divided into 100×100 squares with a size of 235×235 squared meters each cell, and a period of 62 days of communication record details (CDRs) was collected within the area. The original CDRs were aggregated in a time frequency of 10-minutes, and we resample these CDRs with an hour interval for demonstration.

Fig. 1a depicts the hourly aggregated traffic data of a specific cell within the city area for the first two-weeks of November 2013. It can be observed that all the traffic data distinctly follow a seasonal pattern. The daily or weekly traffic of a particular cell is correlated and varying in a similar way, which shows the "temporal correlation" of MTS. Besides, we

also present the city-wide heatmap of internet activities in Fig. 1b. It can be found that the cellular traffic data collected at the neighboring cells varies according to their spatial distribution. Traffic data collected on the same zone may vary in a similar way over time, which presents the "spatial correlation" of MTS.

C. Attention Mechanism

To capture long-range dependency structures of MTS, the attention mechanism was first introduced for machine translation in [21], and has now become a primary concept in deep learning literature. In general, the attention mechanism is a query-key-value model, and it usually adopts scaled dot-product to compute the temporal similarities between queries and keys [19]. The results are given by the normalized weighted sum of training values. Mathematically, its general form can be described as

$$\mathtt{A}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\left[\mathtt{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{T}}{\sqrt{D_{k}}}\right)\right]\mathbf{V}, \tag{3}$$

where $\mathbf{Q}\in\mathbb{R}^{N\times D_k}$, $\mathbf{K}\in\mathbb{R}^{M\times D_k}$ and $\mathbf{V}\in\mathbb{R}^{M\times D_v}$ are queries and training key-value pairs with lengths of N and M, respectively. $(\cdot)^T$ denotes the matrix transpose, and the $\mathtt{softmax}\left(\cdot\right)$ function computes the normalized weights on the last axis of the input tensor. The attention mechanism has been extensively adopted in various ST-MTS prediction tasks, and successful applications include mobile traffic prediction [4], [5] and wireless channel prediction [2], [3]. All these recent researches have adequately verified the superiority of the attention mechanism on ST-MTS prediction.

III. The Proposed Crossover Attention Mechanism

Despite the tremendous success of the attention mechanism in ST-MTS prediction, it does not efficiently exploit cross-domain correlations. In this section, we introduce a simple but effective variant to further enhance attention models' capabilities in ST-MTS prediction tasks.

A. Querying by Temporal Correlation

The vanilla attention mechanism described in (3) can be interpreted as Naradaya-Watson's regression model [19]. Let the temporal view of queries, keys and values be denoted as $\mathbf{Q}=\{\mathbf{q}_i\}_{i=1}^{N}$, $\mathbf{K}=\{\mathbf{k}_i\}_{i=1}^{M}$ and $\mathbf{V}=\{\mathbf{v}_i\}_{i=1}^{M}$ with $\mathbf{q}_i\in\mathbb{R}^{1\times D_k}$, $\mathbf{k}_i\in\mathbb{R}^{1\times D_k}$ and $\mathbf{v}_i\in\mathbb{R}^{1\times D_v}$ being the corresponding spatial vectors. Then, we have the Naradaya-Watson regression model as

$$\mathbf{a}_i=\sum_{j=1}^{M}\sigma\left(\mathbf{q}_i,\mathbf{k}_j\right)\mathbf{v}_j,\quad\forall i=1,2,\ldots,N \tag{4}$$

where $\sigma\left(\cdot\right)$ is a scalar similarity kernel.

Note that in the context of the time-series prediction problem described in (1), key-value pairs can be regarded as the historical covariate-target (features-label) pairs, i.e., $(\mathbf{K},\mathbf{V})\triangleq(\mathbf{X}_{1:t},\mathbf{Z}_{1:t})$. In addition, queries can be regarded as the future h-step covariates, i.e., $\mathbf{Q}\triangleq\mathbf{X}_{t+1:t+h}$. The training space comprises all the observed key-value pairs $\mathcal{D}=\{(\mathbf{k}_i,\mathbf{v}_i)\}_{i=1}^{M}$, and (4) predicts the targets by projecting each $\mathbf{q}$ in $\mathcal{D}$ with the similarity kernel $\sigma\left(\mathbf{q},\mathbf{k}\right)$.

For the attention mechanism in (3), the scaled dot-product similarity between vectors is served as the similarity kernel, which is

$$\sigma\left(\mathbf{q}_i,\mathbf{k}_j\right)=\mathtt{softmax}\left(\frac{\mathbf{q}_i\mathbf{k}_j^{T}}{\sqrt{D_k}}\right). \tag{5}$$

This similarity kernel simply produce the sample cross-covariance matrix $\mathbf{C}=\mathbf{Q}\mathbf{K}^T\in\mathbb{R}^{N\times M}$ between $\mathbf{Q}$ and $\mathbf{K}$, in which the (n,m)-th element $C_{n,m}=\mathrm{Cov}(\mathbf{q}_n,\mathbf{k}_m)$ represents the covariance between the n-th query and the m-th key in $\mathcal{D}$. Therefore, the resulting attention is calculated in the temporal domain. In particular, the sample temporal correlation coefficients are explicitly calculated in the self-attention $\mathtt{A}\left(\mathbf{Z}_{1:t},\mathbf{Z}_{1:t},\mathbf{Z}_{1:t}\right)$. Thus, we refer $\mathtt{A}(\mathbf{Q},\mathbf{K},\mathbf{V})$ as the temporal attention *querying by temporal correlations*.

B. Querying by Spatial Correlation

Inspired by the above analysis, we hereby denote the spatial view of queries and key-value pairs as $\mathbf{Q}=\{\bar{\mathbf{q}}_i\}_{i=1}^{D_k}$, $\mathbf{K}=\{\bar{\mathbf{k}}_i\}_{i=1}^{D_k}$ and $\mathbf{V}=\{\bar{\mathbf{v}}_i\}_{i=1}^{D_v}$ with $\bar{\mathbf{q}}_i\in\mathbb{R}^{N\times 1}$, $\bar{\mathbf{k}}_i\in\mathbb{R}^{M\times 1}$ and $\bar{\mathbf{v}}_i\in\mathbb{R}^{M\times 1}$ being the corresponding temporal vectors. Then, the regression model can be modified as

$$\mathbf{s}_i=\sum_{j=1}^{D_k}\sigma\left(\bar{\mathbf{v}}_i,\bar{\mathbf{k}}_j\right)\bar{\mathbf{q}}_i,\quad\forall i=1,2,\ldots,D_v \tag{6}$$

with a scalar similarity kernel being the scaled dot-product

$$\sigma\left(\bar{\mathbf{v}}_i,\bar{\mathbf{k}}_j\right)=\mathtt{softmax}\left(\frac{\bar{\mathbf{k}}_j^{T}\bar{\mathbf{v}}_i}{\sqrt{M}}\right). \tag{7}$$

When implementing this regression model as a differentiable neural network layer, it can be expressed as

$$\mathtt{S}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\mathbf{Q}\left[\mathtt{softmax}\left(\frac{\mathbf{K}^{T}\mathbf{V}}{\sqrt{M}}\right)\right]. \tag{8}$$

It is clear that the similarity kernel in (8) is computed in the spatial domain. This similarity kernel simply produce the sample cross-covariance matrix $\mathbf{G}=\mathbf{K}^T\mathbf{V}\in\mathbb{R}^{D_k\times D_v}$ between $\mathbf{K}$ and $\mathbf{V}$, in which the (k,v)-th element $C_{k,v}=\mathrm{Cov}(\bar{\mathbf{k}}_k,\bar{\mathbf{v}}_v)$ represents the covariance between the k-th spatial position of keys and the v-th spatial position of values in $\mathcal{D}$. In particular, the sample spatial correlation coefficients are explicitly calculated in the self-attention $\mathtt{S}\left(\mathbf{Z}_{1:t},\mathbf{Z}_{1:t},\mathbf{Z}_{1:t}\right)$. Therefore, we refer $\mathtt{S}(\mathbf{Q},\mathbf{K},\mathbf{V})$ as the spatial attention *querying by spatial correlations*.

C. Crossover Attention

In the neural network implementation, the querying results are controlled by the similarity kernel $\sigma(\cdot)$, and the kernel is implicitly learned via the projections $\mathbf{Q}=\mathbf{I}\mathbf{W}^q$, $\mathbf{K}=\mathbf{I}\mathbf{W}^k$ and $\mathbf{V}=\mathbf{I}\mathbf{W}^v$, where $\mathbf{I}$ denotes the input to the network layer, and $\mathbf{W}^q$, $\mathbf{W}^k$ and $\mathbf{W}^v$ are learnable projection matrices. To fully utilize the spatio-temporal dependency of the input sequence, the results of querying from the temporal attention

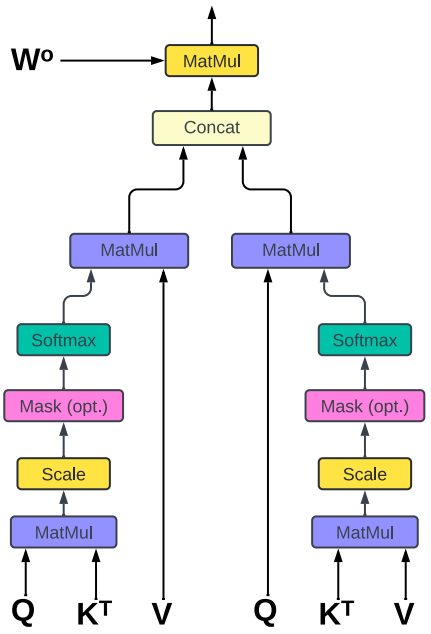


Fig. 2: Computation graph of the proposed crossover attention mechanism.

and the proposed spatial attention should be integrated. With this consideration in mind, we design the crossover attention as

$$\texttt{XOA}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\left[\texttt{A}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right),\texttt{S}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)\right]\mathbf{W}^{O}, \quad (9)$$

where $\mathbf{W}^{O}\in\mathbb{R}^{2D_v\times D_v}$ is utilized to integrate the attention values computed by temporal and spatial correlations. The network layer structure of the proposed crossover attention mechanism is shown in Fig. 2. Note that a mask layer is optional before the `softmax` operation if there is some entries should be masked out for weight computing. The intuition behind the crossover attention design is rather simple: the integration of the cross-domain attentions could help the neural network to learn a more powerful and expressive regression kernel $\sigma(\cdot)$ which jointly considers the temporal and spatial dependencies. As we will show in the experiment results, the proposed crossover attention outperforms the vanilla attention mechanism, and achieves much better prediction performances on different Transformers and datasets.

D. Complexity Analysis

The computational complexity analysis of crossover attention is trivial and straightforward. For the crossover attention described in (9), its computational complexity is determined by both the temporal attention and the spatial attention. For the temporal attention in (3), it computes the temporal correlation with a matrix multiplication between a $N\times D_k$ matrix and a $D_k\times M$ matrix, then the ouput $N\times M$ matrix is multiplied by a $M\times D_v$ matrix. Hence, (3) has a computational complexity of $\mathcal{O}\left(NM(D_k+D_v)\right)$ in general. In particular, it has a computational complexity of $\mathcal{O}\left(T^2D_z\right)$ for the self-attention $\texttt{A}\left(\mathbf{Z}_{1:t},\mathbf{Z}_{1:t},\mathbf{Z}_{1:t}\right)$. Similarly, for the spatial attention in (8), it has a computational complexity of $\mathcal{O}\left(D_kD_v(N+M)\right)$ in general, and particularly the computational complexity for the self-attention $\texttt{S}\left(\mathbf{Z}_{1:t},\mathbf{Z}_{1:t},\mathbf{Z}_{1:t}\right)$ is given by $\mathcal{O}\left(TD_z^2\right)$. Therefore, the total computational complexity of (9) is given by $\mathcal{O}\left(D_kD_v(N+M)+NM(D_k+D_v)\right)$ for the general situation, and the self-crossover-attention $\texttt{XOA}\left(\mathbf{Z}_{1:t},\mathbf{Z}_{1:t},\mathbf{Z}_{1:t}\right)$ has a computational complexity of $\mathcal{O}\left(TD_z(T+D_z)\right)$ in particular. Additionally, crossover attention uses the same set of queries, keys, and values for both temporal and spatial attention sub-modules, making it easy to replace the vanilla attention layer with the crossover attention layer.

TABLE I: Performance Comparisons for the Milan Dataset.

Model	MAE	NRMSE	R^2
HA	18.7226	0.9687	0.4419
ARIMA	17.1895	0.8813	0.6564
LSTM [11]	13.9438	0.6079	0.7802
STDenseNet [10]	12.3168	0.6442	0.7842
ConvLSTM [17]	11.2308	0.5652	0.8097
ST-Tran [2]	10.3820	0.5521	0.8187
ST-Tran-XOA	**9.9943**	**0.5508**	**0.8196**

IV. Performance Evaluation

In this section, we will show numerical results to verify the effectiveness of the proposed crossover attention. The experiments are conducted by replacing the attention layers inside of recent developed Transformers with our proposed crossover attention layer. Since the replacement does not change the model's size, clear performance comparisons can be present for verification.

A. Environment Setup

In the experiments, we carry out simulations on two different ST-MTS prediction datasets listed below,

- **Milan**: This dataset contains the 62 days' cellular mobile traffic data for the city Milan in Italy [20]. We have introduced and illustrated some details of this dataset in Sec. II-B. The whole city area is divided into 100×100 square cells, and the cellular traffic data is resampled in hourly granularity. Note that we will focus on predicting the number of call-ins for the selected 20×20 cells for this dataset in the experiments.
- **SanDiego**: By comparing to *Milan*, this dataset is a much larger one, and it is also a subset of the LargeST benchmark dataset [6]. The dataset contains the road network traffic data for over $17,000$ road segments and 700 sensors in the area around the city San Diego, USA. The data is recorded in 5-minutes interval for five years from 2017 to 2021. Note that we will focus on predicting the traffic volumes for all the sensors of this dataset in the experiments.

B. Competing Algorithms

As we have mentioned, our experiments are conducted by replacing attention modules inside of two recent developed Transformers for ST-MTS prediction, where the details of

TABLE II: Performance Comparisons for the SanDiego Dataset.

Model	Horizon 3			Horizon 6			Horizon 12			Average		
	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE
HL	33.61	50.97	20.77%	57.8	84.92	37.73%	101.74	140.14	76.84%	60.79	87.4	41.88%
LSTM [11]	19.03	30.53	11.81%	25.84	40.87	16.44%	37.63	59.07	25.45%	26.44	41.73	17.20%
DCRNN [22]	17.14	27.47	11.12%	20.99	33.29	13.95%	26.99	42.86	18.67%	21.03	33.37	14.13%
AGCRN [23]	15.71	27.85	11.48%	18.06	31.51	13.06%	21.86	39.44	16.52%	18.09	32.01	13.28%
STGCN [7]	17.45	29.99	12.42%	19.55	33.69	13.68%	23.21	41.23	16.32%	19.67	34.14	13.86%
STTN [16]	16.22	26.22	10.63%	18.76	30.98	12.80%	22.62	39.09	16.14%	18.69	31.11	12.82%
STTN-XOA	**15.57**	**25.48**	**10.01%**	**17.87**	**29.78**	**11.74%**	**21.76**	**37.19**	**14.81%**	**17.85**	**29.74**	**11.72%**

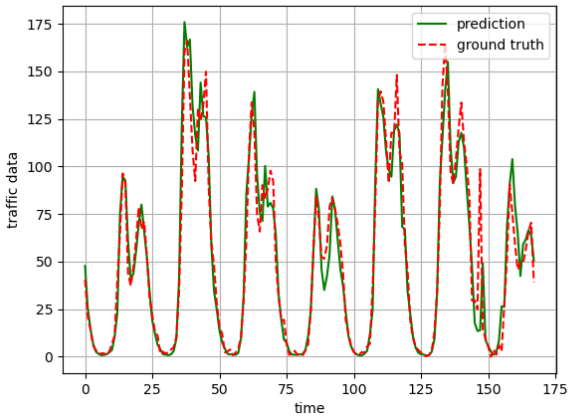


Fig. 3: The fitness curve of ST-Tran-XOA for the cellular traffic flows in Milan.

these Transformers could be found in [2], [16] and [6]. For the ease of reader's convenience, we list below the abbreviations of the competing algorithms or neural networks,

- **HA**: The historical average (HA) algorithm which takes the average of its history as the prediction result.
- **HL**: The Historical Last (HL) which simply uses the last observation as the future prediction.
- **ARIMA**: The well-known autoregressive integrated moving average (ARIMA) algorithm that is implemented by the *statsmodels* python library.
- **LSTM**: The long-short term memory (LSTM) neural network for time-series prediction [11].
- **ConvLSTM**: The convolutional LSTM which is proposed for ST-MTS prediction in [17].
- **STDenseNet**: STDenseNet [10] is a prediction model which learns spatio-temporal dependency structure using densely connected CNNs.
- **DCRNN**: The diffusion convolutional recurrent neural network (DCRNN) proposed in [22].
- **AGCRN**: The adaptive graph convolutional recurrent network (AGCRN) proposed in [23].
- **STGCN**: The spatio-temporal graph convolutional networks proposed in [7].
- **ST-Tran**: ST-Tran is an encoder-decoder Transformer designed for ST-MTS prediction [2].
- **STTN**: The spatio-temporal Transformer networks (STTN) for the spatio-temporal traffic prediction [16]. It not only uses the attention mechanism, but also integrates GCNs inside the model.
- **ST-Tran-XOA**: Our modified version of ST-Tran, where the attention layers are replaced by our proposed crossover attention layers.
- **STTN-XOA**: Our modified version of STTN, where the attention layers are replaced by our proposed crossover attention layers.

Note that the simulations of ST-Tran-XOA and STTN-XOA are all performed following the same random seeds, hyper-parameters, instructions and datasets as described in the original papers. By this way, we can present a very clear and straightforward performance comparison results to show the effectiveness of the proposed crossover attention mechanism.

To compare the prediction performance, we adopt several metrics for evaluating the prediction accuracy of each models, including mean absolute error (MAE), mean absolute percentage error (MAPE), mean square error (MSE), normalized root mean square errors (NRMSE), and R-squared coefficient of determination (R^2). Note that R^2 provides information about the goodness of a fitting model, and its value normally varies within $[0, 1]$ with 1 indicating perfect fitting.

C. Numerical Result and Discussion

Table. I summarizes the prediction performance comparisons of the competing models for the *Milan* dataset. We can observe from Table. I that our proposed crossover attention achieves the best prediction accuracy in terms of MAE, NRMSE, and R^2. It can be found that ST-Tran-XOA achieves the highest R^2 score among the seven competing models, which indicates that ST-Tran-XOA learns a most fitting model for the spatio-temporal cellular traffic data in the *Milan* dataset. By comparing ST-Tran-XOA with ST-Tran, we can conclude that the proposed crossover attention mechanism can help the model exploit the spatio-temporal dependencies of data and achieve lower prediction errors. In addition to the numerical results presented in Table. I, we also depict the predicted results in Fig. 3 to illustrate the model fitness of ST-Tran-XOA. It can be observe from the figure that the ST-Tran-XOA can accurately and smoothly predict the trends and values of the future mobile traffic data of a specific cell within the city area, which clearly verifies the effectiveness of the proposed crossover attention mechanism.

In addition to the *Milan* dataset, Table. II illustrates the prediction results of the aforementioned prediction models for

the *SanDiego* dataset. It can also be found from Table. II that our proposed crossover attention mechanism outperforms the competing models in terms of prediction errors. More specifically, we can observe that STTN-XOA achieves the lowest prediction errors among the seven competing models for the predition horizons from 3 to 12. It's important to highlight that this improvement is substantial as it surpasses the performance of GNN-based models, where the vanilla attention based STTN model fails to do so. This shows that the proposed crossover attention can also help the model in multi-step prediction tasks. By comparing STTN-XOA with STTN, we can conclude that the Transformer model's ability of capturing spatio-temporal dependencies is significantly enhanced by the proposed crossover attention. These results further demonstrate the effectiveness of the proposed crossover attention mechanism.

V. Conclusion

In this paper, we have investigated the MTS prediction problem with a focus on predicting spatio-temporal correlated traffic flows such as mobile traffic and road traffic. To improve the prediction accuracy of the existing attention-based models, we designed a simple-but-effective crossover attention mechanism to help the network understand the spatio-temporal patterns of input data. Experiment results on realistic datasets clearly verified the effectiveness of our proposed crossover attention mechanism. Since the proposed crossover attention can be freely and seamlessly replaced in many Tranformers, we believe that it can serve as an effective fundamental building blocks in Transformer models, providing brand new insights for the future model design.

Acknowledgement

This work was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant references FNR/C19/IS/13718904/ASWELL and FNR/C22/IS/17220888/RUTINE.

References

[1] S. Mehrizi and S. Chatzinotas, "Network traffic modeling and prediction using graph gaussian processes," *IEEE Access*, vol. 10, pp. 132 644–132 655, 2022.

[2] Q. Liu, J. Li, and Z. Lu, "ST-Tran: Spatial-temporal transformer for cellular traffic prediction," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3325–3329, 2021.

[3] X. Wang, Z. Wang, K. Yang, Z. Song, C. Bian, J. Feng, and C. Deng, "A survey on deep learning for cellular traffic prediction," *Intelligent Computing*, vol. 3, p. 0054, 2024.

[4] C. Wu, X. Yi, Y. Zhu, W. Wang, L. You, and X. Gao, "Channel prediction in high-mobility massive MIMO: from spatio-temporal autoregression to deep learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1915–1930, 2021.

[5] M. K. Shehzad, L. Rose, S. Wesemann, and M. Assaad, "ML-based massive MIMO channel prediction: Does it work on real-world data?" *IEEE Wirel. Commun. Lett.*, vol. 11, no. 4, pp. 811–815, 2022.

[6] X. Liu, Y. Xia, Y. Liang, J. Hu, Y. Wang, L. Bai, C. Huang, Z. Liu, B. Hooi, and R. Zimmermann, "LargeST: A benchmark dataset for large-scale traffic forecasting," in *Proc. NeurIPS*, 2023.

[7] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. IJCAI*, 2018, pp. 3634–3640.

[8] F. Chiariotti, M. Drago, P. Testolina, M. Lecci, A. Zanella, and M. Zorzi, "Temporal characterization and prediction of VR traffic: A network slicing use case," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 3890–3908, 2024.

[9] K. He, T. X. Vu, D. T. Hoang, D. N. Nguyen, S. Chatzinotas, and B. Ottersten, "Risk-aware antenna selection for multiuser massive MIMO under incomplete CSI," *IEEE Trans. Wirel. Commun.*, pp. 1–1, 2024.

[10] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656–1659, 2018.

[11] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 4, pp. 554–557, 2018.

[12] F. Sun, P. Wang, J. Zhao, N. Xu, J. Zeng, J. Tao, K. Song, C. Deng, J. C. S. Lui, and X. Guan, "Mobile data traffic prediction by exploiting time-evolving user mobility patterns," *IEEE Trans. Mob. Comput.*, vol. 21, no. 12, pp. 4456–4470, 2022.

[13] T. Qi, G. Li, L. Chen, and Y. Xue, "ADGCN: an asynchronous dilation graph convolutional network for traffic flow prediction," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 4001–4014, 2022.

[14] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101819, 2023.

[15] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," in *Proc. IJCAI*, 2023, pp. 6778–6786.

[16] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.-J. Qi, and H. Xiong, "Spatial-temporal transformer networks for traffic flow forecasting," *arXiv preprint arXiv:2001.02908*, 2020.

[17] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. NIPS*, 2015, pp. 802–810.

[18] G. Liu, Z. Hu, L. Wang, J. Xue, H. Yin, and D. Gesbert, "Spatio-temporal neural network for channel prediction in massive MIMO-OFDM systems," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 8003–8016, 2022.

[19] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 5, pp. 53:1–53:32, 2021.

[20] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of Milan and the province of Trentino," *Scientific Data*, vol. 2, no. 1, pp. 1–15, 2015.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[22] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. ICLR*, 2018.

[23] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. NIPS*, 2020.