2023 IEEE International Conference on Data Mining (ICDM) | 979-8-3503-0788-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDM58522.2023.00025

Boosting Urban Prediction via Addressing Spatial-Temporal Distribution Shift

Xuanming Hu[1*†], Wei Fan[2†], Kun Yi[3], Pengfei Wang[4], Yuanbo Xu[5], Yanjie Fu[1], Pengyang Wang[2‡]
[1] School of Computing and Augmented Intelligence, Arizona State University, Tempe, USA
[2] Department of CIS, SKL-IOTSC, University of Macau, Macau S.A.R., China
[3] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[4] Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
[5] College of Computer Science and Technology, Jilin University, Changchun, China
{solomonhxm, yanjie.fu}@asu.edu, {weifan, pywang}@um.edu.mo, yikun@bit.edu.cn,
pfwang@cnic.cn, yuanbox@jlu.edu.cn

Abstract*—Urban prediction tasks that aim to model the complicated spatial and temporal patterns of urban indicators (such as weather, vehicle charging demand, etc.) for accurate prediction, have been increasingly important in constructing smart cities and accelerating the urbanization process in the modern era. However, most existing works of urban prediction have only concentrated on spatial and temporal *correlations*, but ignored the effect of *distribution shift* from spatial and temporal perspectives; this could largely hinder the performance of urban prediction tasks. In order to solve this problem, in this paper, we propose a Shift-Aware Urban Prediction (SAUP) framework to eliminate the inherent shift effect among spatial-temporal urban time series data. Specifically, SAUP starts with a Shift Elimination Module, built upon our proposed Spatial-Temporal Attention Flows (STAF) composed of invertible attentions and coupling layers of normalizing flows in order to transform the raw shifted data into a unified distribution to remove the spatiotemporal shift. After the shift effect is eliminated, the Correlation Processing Module of SAUP further captures the core correlations to learn spatiotemporal dependencies, in which topological correlations and geographic correlations are jointly learned by GCN and CNN based on pre-defined graphs and extracted POI information. In addition, SAUP includes a model-agnostic Forecasting Module, which can be employed as any forecasting architecture to accomplish the predictions. To recover the raw distribution information, the output of the Forecasting Module is further taken for the inverse transformation of the Shift Elimination Module to produce the final forecasts. We have conducted extensive experiments in the SAUP framework, coupled with six state-of-the-art spatiotemporal forecasting models on two real-world datasets. Experimental results have demonstrated the consistent improvements of SAUP over the baseline algorithms.

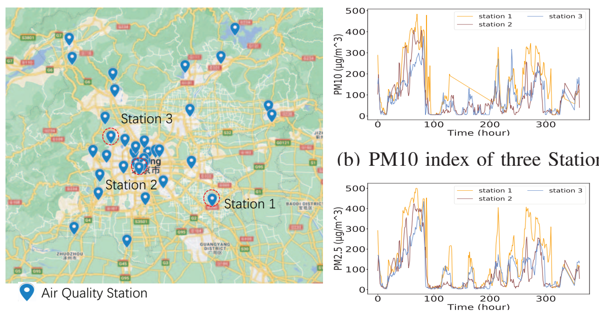
I. Introduction

The high-speed urbanization process in the modern era brings great challenges to public administration due to the rapidly growing urban construction demands [1]. Under such a situation, urban prediction has become an appealing research application recently [2]–[4], showing its great potential to construct smart cities. [5]. As well-known as spatial-temporal forecasting, urban prediction focuses on predicting the future status (e.g., traffic flow, air quality, temperature, etc.) of the

[*]This work was done when the first author was an undergraduate research assistant in SKL-IOTSC at the University of Macau.

[†]Both authors contributed equally to this research.

[‡]Corresponding author



(a) Spatial distribution of air quality stations (b) PM10 index of three Stations (c) PM2.5 index of three Stations

Fig. 1. An illustration of data distribution collected from three stations. These series share correlated patterns but also have the spatial-temporal shift.

urban area based on the historical status, which is universally conducted in city service systems (e.g., transportation system, weather forecasting system, etc). Nevertheless, on account of the intricate intra-dependencies (i.e., correlations within one single node in different time periods) and inter-dependencies (i.e., correlations among numerous time series from different nodes) of urban spatial-temporal series, urban predication can be an extremely challenging task [6].

Previously, the prevailing methods of urban prediction are to take advantage of statistical methods such as Auto-Regressive Integrated Moving Average (ARIMA) [7] and Vector Auto-Regression (VAR) [8]. These methods, however, are only fit for simple linear temporal relations and would fail to make precise predictions for more complicated spatial-temporal dependencies due to the uncertain and dynamic urban environment [6]. With the development of deep learning techniques, researchers turn to construct deep neural network architectures to discover the hidden correlation shared by urban big data [9]–[14]. To model the non-euclidean structure of the graph and capture the spatial correlation, they generally utilize GCN-based model [3], [6], [15]. As for the purpose of modeling temporal dependencies, they employ recurrent neural networks or temporal convolution modules as their tools [10], [16], [17].

Though existing works concentrated on modeling complicated spatial and temporal dependencies [18], they ignore the ubiquitous shift effect among urban data. To illustrate it, we randomly choose three air quality stations in Beijing and plot the 360h-records of PM2.5 and PM10 in Figure 1 (all of the missing values are filled by linear interpolation). The figure demonstrates an appealing fact that there is still much difference though diverse stations and different time periods share some similar patterns. To describe these two kinds of differences in spatial and temporal domains, we introduce the item, ”shift” effect and classify the shift into two types by their causes, i.e., spatial shift effect and temporal shift effect. Investigating the cause and influence of shift effects can be essential for further research. First, the temporal shift effect is important in urban prediction. Most urban prediction models assume the temporal patterns of historical values and future values are stationary; however, this assumption is not realistic, as the joint distribution of real-world time series data is always shifted, leading to the ”temporal shift” effect [19], [20] which interferes urban prediction models to generate the prediction. Hence, it is essential to mitigate the temporal shift effect to enhance the capture of dependencies. Furthermore, the spatial shift effect is also important. To elaborate on the spatial shift, we take the air quality stations of Beijing (see Figure 1) as an example. Although there is a strong correlation among different stations, we can also find out that different series also indicate evident dissimilarity. It is not strenuous to see that different series have time lags and different mean as well as variance. Concretely, due to the diverse geographic locations, different stations have dissimilar distributions which greatly impedes the spatial-temporal forecasting models to capture spatial correlation. In this paper, we use the term ”spatial shift” effect to describe this dissimilarity between nodes in general. Therefore, eliminating spatial effects can be advantageous for the improvement of accuracy. If we do not eliminate the shift effect before capturing the correlation, it is highly possible that forecasting models focus more on heterogeneous information rather than capturing similar patterns and making a biased prediction. To this end, we propose a Shift Aware Urban Prediction (SAUP) framework, which is composed of the Shift Elimination Module to eliminate the shift effect, the Correlation Processing Module to capture the spatio-temporal correlation, and the Forecasting Module to accomplish the prediction task.

Our main question is how can we eliminate the shift effect including spatial shift and temporal shift effect. Since the spatial shift effect represents the dissimilarity between nodes and the temporal shift effect means the dissimilarity of different time points, we can regard their data distributions as heterogeneous distributions. Thus an intuitive method to remove shift effects is to transform them into one unified distribution which is favorable for forecasting model before capturing their dependencies. To this end, we propose Spatial-Temporal Attention Flow (STAF) as our Shift Elimination Module for the distribution learning to eliminate the spatial-temporal shift effect, enlightened by the remarkable distribution transformation ability of normalizing flows [21]. In addition, we notice flow-based models are invertible neural networks [22], which not only can transform distributions but also can recover shift information. We make use of this advantage and make our elimination process also reversible, in order to make the forecasting conducted in the transformed stationary space and then distribution information recovered by the inverse transformation of STAF. STAF is built upon the coupling layers [23], [24] widely used in many normalizing flows [24]. Also, the shift always exists consistently and has a long-term effect, we need to include a structure to capture the global relationship of the shift effect. Since we need to inverse the output back to the original distribution, this structure should also be invertible. Thus we apply invertible attention [25] in our STAF to enhance the distribution learning, which is totally invertible, easy to implement, and performs well to capture the long-term relationship [25].

Our another question is how to capture spatial temporal correlation in urban prediction while the shift effect exists. Existing works always disregard the influence of the shift effect and directly capture spatial correlations [3], [14], [15]. However, it is hard for these models to capture the real correlation among the series due to the existence of the shift effect. As a result, we aim to first eliminate the shift effect and then capture spatial temporal dependencies in our Correlation Processing Module, in order to accurately model the patterns of the spatial-temporal urban series. Specifically, to better capture spatial temporal dependencies in the transformed space of STAF, we consider and fuse two kinds of correlations, namely topological correlation and geographic correlation to conduct dependency learning. For the topological correlation, we employ a Graph Convolutional Network (GCN) to achieve the desired effect to model the graph structure data [26], which models the complex dependencies with the pre-defined adjacency matrix. For the geographic correlation, we consider Point of interest (POI) information in grid-level and utilize Convolutional Neural Network (CNN) to learn the embeddings, which plays a crucial role in spatial-temporal forecasting [27]. The major contributions of this work can be briefly summarized as:

- We introduce the shift effect to measure the dissimilarity between different nodes and different time periods, and we propose the Shift-Aware Urban Prediction (SAUP) framework to convert the original data into a stationary space for prediction and capture topological correlation and geographic correlation.
- We design an invertible Spatial-Temporal Attention Flows (STAF) built upon coupling layers and invertible attention to eliminate both temporal and spatial shifts of urban data.
- We construct a Correlation Processing Module to capture topological correlation and geographic correlation among spatial-temporal series and fuse them together after the shift effect is removed.
- We conduct several experiments on two real-world datasets with six state-of-the-art forecasting models as baselines. Extensive empirical results indicate that our proposed framework can consistently improve the prediction performance over the baseline algorithms.

II. PRELIMINARIES

A. Problem Definition

In this paper, we concentrate on solving the multi-step urban prediction problem. Given a certain urban prediction task (e.g., traffic prediction [6], weather prediction [28]) that contains multiple correlated time series represented as $\{\mathbf{X}_1^{(1:N)}, \mathbf{X}_2^{(1:N)}, \cdots, \mathbf{X}_t^{(1:N)}, \cdots\}$, we let $\mathbf{X}_t^{(1:N)} = \{\mathbf{x}_t^{(1:N),1}, \mathbf{x}_t^{(1:N),2}, \cdots, \mathbf{x}_t^{(1:N),D}\} \in \mathbb{R}^{N\times D}$ as the recording of N sources of D dimensions of urban information (e.g. vehicle speed, weather, and charging demand) at time-step t. Our target is to make predictions of the future values of the correlated series based on the observed historical values. We formulate the problem as finding a mapping function $\mathcal{F}_\theta$ to forecast the next τ time-steps spatial-temporal data based on the past T steps historical data (T is defined as the length look-back window):

$$\mathbf{X}_{t+1}^{(1:N)}, \mathbf{X}_{t+2}^{(1:N)}, ..., \mathbf{X}_{t+\tau}^{(1:N)} = \mathcal{F}_\theta(\mathbf{X}_t^{(1:N)}, \mathbf{X}_{t-1}^{(1:N)}, ..., \mathbf{X}_{t-T+1}^{(1:N)}) \tag{1}$$

where θ is denoted as all the learnable parameters in the model.

For the purpose of investigating the spatial correlation between diverse urban series, we introduce topological and geographic knowledge as parts of the model input. The urban forecasting task can be further formulated on graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ as topological information, where $\mathcal{V}$ denotes the set of nodes in a traffic graph, and $|\mathcal{V}| = N$, $\mathcal{E}$ represents the set of edges between nodes, and $\mathcal{A} \in \mathbb{R}^{N\times N}$ is the adjacent matrix of the graph. If $v_i, v_j \in \mathcal{V}$ and $(v_i, v_j) \in \mathcal{E}$, $\mathcal{A}_{ij} = 1$, otherwise $\mathcal{A}_{ij} = 0$. In addition, we would like to include the POI as geographic information. We denote $\mathcal{P} \in \mathbb{R}^{C\times N}$, where N is the number of nodes and C is the number of categories of POI. Thus, the problem can be further adjusted as:

$$\begin{aligned}&\mathbf{X}_{t+1}^{(1:N)}, \mathbf{X}_{t+2}^{(1:N)}, ..., \mathbf{X}_{t+\tau}^{(1:N)} = \\ &\quad \mathcal{F}_\theta(\mathbf{X}_t^{(1:N)}, \mathbf{X}_{t-1}^{(1:N)}, ..., \mathbf{X}_{t-T+1}^{(1:N)}; \mathcal{G}; \mathcal{P})\end{aligned} \tag{2}$$

We define $\mathbf{X}_{t-T:t}^{(1:N)} = \{\mathbf{X}_{t-T+1}^{(1:N)}, \mathbf{X}_{t-T+2}^{(1:N)}, ..., \mathbf{X}_{t-1}^{(1:N)}\}$ as the lookback window for brevity in the following sections of this paper, accordingly we write $\mathbf{X}_{t:t+\tau}^{(1:N)}$ as future values needing to predict.

III. METHODOLOGY

In this section, we first introduce the architecture of our general framework, i.e. the Shift-Aware Urban Prediction framework (SAUP). Next, we delineate three modules of our framework which collaborate to model the complex spatial-temporal patterns in Section III-A, the Shift Elimination Module in Section III-B, the Correlation Processing Module in Section III-C, the Forecasting Module and the inverse transformation of STAF in Section III-D.

A. Framework Overview

Figure 2 shows the structure of our proposed Shift-Aware Urban Prediction framework, including (1) To eliminate the shift effect, we design Shift Elimination Module with Spatial-Temporal Attention Flows (STAF) to transfer the original distribution into a hidden stationary distribution which is favorable for the following modules to model the complex spatial-temporal relationship. (2) After eliminating the shift effect, we utilize a Correlation Processing Module to extract the hidden patterns of the pre-defined graph and POI information in order to capture the correlation among spatial-temporal data. (3) And we use the forecasting module that is suitable for any spatial-temporal forecasting models to make a prediction. (4) Ultimately, we devise the inverse transformation of the Shift Elimination Module to inverse the output of the forecasting module back to the original space and obtain the final results, so the shift information can be recovered.

B. Shift Elimination Module

Shift exists ubiquitously among urban spatial-temporal series due to the dynamic urban environment. However, most recent works only have demonstrated that spatial-temporal correlation plays a crucial role in urban prediction task [18] and they ignore the effect of shift which is also essential in spatial-temporal forecasting. Inspired by [19], our core thought to deal with the shift effect can be split into three steps. First of all, we remove the non-stationary shift effect of the series. Then we utilize the series without the shift effect to forecast. So that the non-stationary effect can be excluded from the forecasting process. At last, we recover the shift information. To maintain the accuracy and completeness of the information, the process of removing the shift effect has to be totally invertible, i.e., the recovering process has to be the inverse transformation of the removing process. Moreover, as mentioned in the previous section, if we transform the series into a unified distribution, the shift effect can be eliminated. So, in order to eliminate the shift effect, we propose a totally invertible Spatial-Temporal Attention Flows (STAF) as Shift Elimination Module by utilizing invertible attention [25] as well as coupling layers [24] to implement. Such a solution is built upon the Normalizing flows which is famous for its remarkable ability of distribution transformation.

Coupling Layers To eliminate the effect of shift, we have to consider the nature of the shift effect. As mentioned in the previous section, the spatial shift occurs because different nodes share different distributions due to their geographic location, and the temporal shift occurs since time frames have heterogeneous distributions So if we want to eliminate the effect of shift, we need to transfer the original distributions to unified distributions which are stationary and beneficial for the backbones to capture complex patterns among spatial-temporal data. Due to the marvelous ability of Normalizing flows for distribution transformation, we utilize two coupling layers, i.e. Spatial Coupling Layers and Temporal Coupling Layers, to eliminate the shift effect of spatial and temporal dimensions.

Usually, designing tractable transformation is not simple and the inverse $\mathbf{x} = f^{-1}(z)$ is hard to evaluate too. To tackle these problems and to increase the expression of the transformation, we utilize the bijection function proposed by Real NVP [24] which is called the affine coupling layer. First of all, we use Spatial Coupling Layers to eliminate the spatial shift effect.

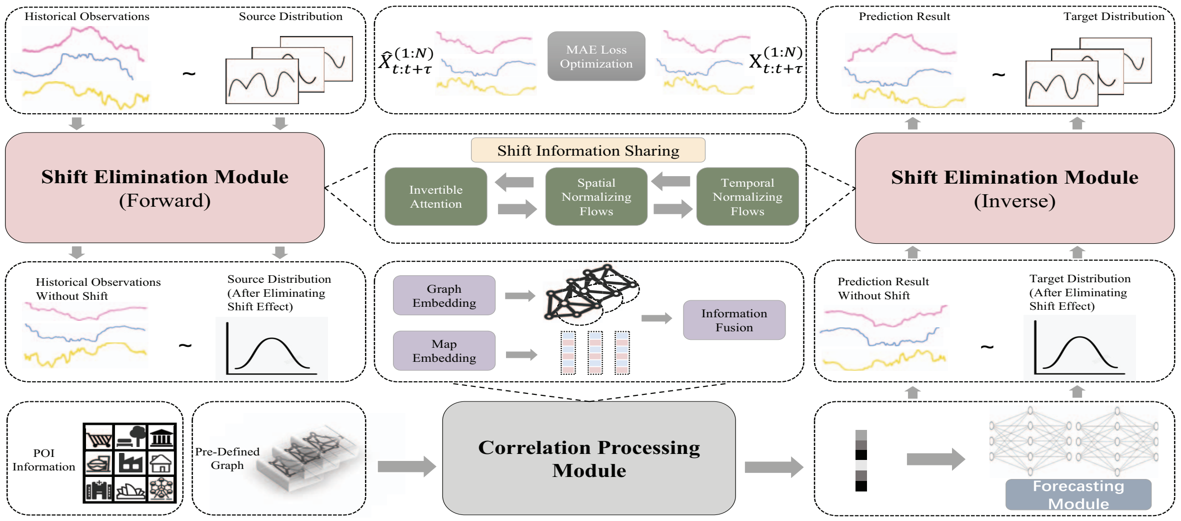

Historical Observations
Source Distribution
MAE Loss Optimization
$\hat{X}^{(1:N)}_{t:t+\tau}$
$X^{(1:N)}_{t:t+\tau}$
Prediction Result
Target Distribution
Shift Elimination Module (Forward)
Shift Information Sharing
Invertible Attention
Spatial Normalizing Flows
Temporal Normalizing Flows
Shift Elimination Module (Inverse)
Historical Observations Without Shift
Source Distribution (After Eliminating Shift Effect)
Graph Embedding
Map Embedding
Information Fusion
Prediction Result Without Shift
Target Distribution (After Eliminating Shift Effect)
POI Information
Pre-Defined Graph
Correlation Processing Module
Forecasting Module

Fig. 2. The framework overview of Spatial-Aware Urban Predication (SAUP).

The input of Spatial Coupling Layers is $\widetilde{\mathbf{X}}^{(1:N)}_{t-T:t} \in \mathbb{R}^{T\times N\times D}$, it has an input space of $\mathcal{X}$, and $n_c < N$ which can be an arbitrary partition, but in this work, we set $n_c = \lceil \frac{D}{2} \rceil$. The output $\mathbf{Z}^{(1:N)}_{t-T:t}$ can be calculated as following:

$$\begin{aligned} \mathbf{Z}^{(1:n_c)}_{t-T:t} &= \widetilde{\mathbf{X}}^{(1:n_c)}_{t-T:t} \\ \mathbf{Z}^{(n_c:N)}_{t-T:t} &= \widetilde{\mathbf{X}}^{(n_c:N)}_{t-T:t} \odot exp(s(\widetilde{\mathbf{X}}^{(1:n_c)}_{t-T:t})) + t(\widetilde{\mathbf{X}}^{(1:n_c)}_{t-T:t}) \end{aligned} \tag{3}$$

Where $\mathbf{Z}^{(1:N)}_{t-T:t} \in \mathbb{R}^{T\times N\times D}$ and $\mathbf{Z}^{(1:N)}_{t-T:t}$ has a hidden space, we denote it as $\mathcal{Z}$.

Then We employ Temporal Coupling Layers to eliminate the temporal shift effect. Utilizing the input $\mathbf{Z}^{(1:N)}_{t-T:t} \in \mathbb{R}^{T\times N\times D}$ and set $t_c = \lceil \frac{T}{2} \rceil$, the output $\widetilde{\mathbf{Z}}^{(1:N)}_{t-T:t} \in \mathbb{R}^{T\times N\times D}$ and $\widetilde{\mathbf{Z}}^{(1:N)}_{t-T:t} \in \mathbb{R}^{T\times N\times D}$ can be formulated as:

$$\begin{aligned} \widetilde{\mathbf{Z}}^{(1:N)}_{t-T:t_c} &= \mathbf{Z}^{(1:N)}_{t-T:t_c} \\ \widetilde{\mathbf{Z}}^{(1:N)}_{t_c:t} &= \mathbf{Z}^{(1:N)}_{t_c:t} \odot exp(s(\mathbf{Z}^{(1:N)}_{t-T:t_c})) + t(\mathbf{Z}^{(1:N)}_{t-T:t_c}) \end{aligned} \tag{4}$$

Where $\widetilde{\mathbf{Z}}^{(1:N)}_{t-T:t} \in \mathbb{R}^{T\times N\times D}$, and the hidden output space of as $\widetilde{\mathbf{Z}}^{(1:N)}_{t-T:t}$ is denoted as $\mathcal{Z}'$.

Invertible Attention Typically, the sensors can cover a vast geographical space and the time periods of the spatial-temporal series are extensive. This can be a challenge for the Shift Elimination Module to cover global shift information. So a non-local structure better captures long-distance dependencies and thus can be a remedy for this problem. From this perspective, we consider self-attention [29] as a useful tool since it is an efficient mechanism to capture non-local dependencies. In this work, we use Non-local Neural Networks proposed by [30]. We denote the input as $\mathbf{X}^{(1:N)}_{t-T:t} \in \mathbb{R}^{T\times N\times D}$. So we can write the attention as:

$$\widetilde{\mathbf{X}}^{(1:N)}_{t_i} = \frac{1}{N(\mathbf{X}^{(1:N)}_{t-T:t})} \sum\nolimits_{\forall j} r(\mathbf{X}^{(1:N)}_{t_i}, \mathbf{X}^{(1:N)}_{t_j}) F(\mathbf{X}^{(1:N)}_{t_j}) \tag{5}$$

where $\widetilde{\mathbf{X}}^{(1:N)}_{t-T:t} = \{\widetilde{\mathbf{X}}^{(1:N)}_{t-T+1}, \widetilde{\mathbf{X}}^{(1:N)}_{t-T+2}, ..., \widetilde{\mathbf{X}}^{(1:N)}_{t-1}\}$ is output and $\widetilde{\mathbf{X}}^{(1:N)}_{t-T:t} \in \mathbb{R}^{T\times N\times D}$, t_i, t_j are denoted as all possible positions' index of T dimension, $F(\cdot)$ is the projection function of the input x to map it to the feature space which is also called feature map, $r(\cdot)$ is the response function and $N(\cdot)$ is a normalizing factor. We denote response map as $R(\mathbf{X}^{(1:N)}_{t-T:t})$ which is the set of $r(\mathbf{X}^{(1:N)}_{t_i}, \mathbf{X}^{(1:N)}_{t_j})$ for all possible pairs of t_i, t_j after normalizing.

Unlike recurrent and convolutional operations, non-local attention can ignore the spatial-temporal distance and better capture the long-range correlation among spatial-temporal series. The response function of Embedded Gaussian Attention can be defined as: $r(\mathbf{X}^{(1:N)}_{t_i}, \mathbf{X}^{(1:N)}_{t_j}) = exp(\theta(\mathbf{X}^{(1:N)}_{t_i})^T \times \phi(\mathbf{X}^{(1:N)}_{t_j}))$ where $\theta(\mathbf{X}^{(1:N)}_{t_i}) = W_\theta \mathbf{X}^{(1:N)}_{t_i}$ and $\phi(\mathbf{X}^{(1:N)}_{t_j}) = W_\phi(\mathbf{X}^{(1:N)}_{t_j})$ are the two embedding. And the normalizing factor $N(\mathbf{X}^{(1:N)}_{t-T:t})$ is $N(\mathbf{X}^{(1:N)}_{t-T:t}) = \sum_{\forall j} r(\mathbf{X}^{(1:N)}_{t_i}, \mathbf{X}^{(1:N)}_{t_j})$.

As mentioned previously, our solution to eliminate shift is to remove the non-stationary shift effects first and recover them after forecasting by the inverse transformation of the Shift Elimination Module. Hence, to ensure the recovering process preserves all the accurate shift information, the structure of the Shift Elimination Module should be invertible. As part of the Shift Elimination Module, the non-local attention should be also invertible. However, most of the attention mechanism is not invertible since their transformation process is not bijective. So we introduce the invertible attention proposed by [25] to solve this problem. They prove that attention can be invertible by restricting the response map, feature mapping F, and utilizing a Lipschitz-constrained convolution at the last step of residual branch. We will discuss the details of constraints in the Theoretical Analysis parts later.

Theoretical Analysis We are going to discuss the theoretical analysis of our Spatial-Temporal Attention Flows here. Starting from the basic mechanism of invertible attention, we will introduce how it can be invertible.

First of all, we introduce the Inversed Residual Structure. Given an input $\mathbf{x}^{(n),d}_{t-T:t} \in \mathbb{R}^T$, we define the residual structure as: $H(\mathbf{x}^{(n),d}_{t-T:t}) = \mathbf{x}^{(n),d}_{t-T:t} + G(\mathbf{x}^{(n),d}_{t-T:t})$ For the propose of making $H(\mathbf{x}^{(n),d}_{t-T:t})$ invertible, we need to constrain the

Algorithm 1 Inverse of Residual Block

```
Input: Output from residual layer $\mathbf{z}_{t-T:t}^{(n),d}$, residual block $G$, number of iterations $n$
1: Initial: $\mathbf{x}_{t-T:t}^{(n),d,0} \leftarrow \mathbf{z}_{t-T:t}^{(n),d}$
2: for i = 0, 1, ..., n do
3:     $\mathbf{x}_{t-T:t}^{(n),d,i+1} \leftarrow \mathbf{z}_{t-T:t}^{(n),d} - G(\mathbf{x}_{t-T:t}^{(n),d,i})$
4: end for
5: return $\mathbf{x}_{t-T:t}^{(n),d,n+1}$
```

Lipschitz constant $\mathbb{L}$ of G [31]. Here we give the Lemma 1 to demonstrate one sufficient condition for the Lipschitz constrain.

Lemma 1 : Given the input $\mathbf{x}_{t-T:t}^{(n),d} \in \mathbb{R}^T$, $H(\mathbf{x}_{t-T:t}^{(n),d})$ is invertible if $\mathbb{L}(G) = c$ where $c \in (0,1)$, and $\mathbb{L}(G)$ is defined as:

$$\mathbb{L}(G) = sup_{\mathbf{x}_{t_1}^{(n),d} \neq \mathbf{x}_{t_2}^{(n),d}} \frac{||G(\mathbf{x}_{t_1}^{(n),d}) - G(\mathbf{x}_{t_2}^{(n),d})||}{||\mathbf{x}_{t_1}^{(n),d} - \mathbf{x}_{t_2}^{(n),d}||} \tag{6}$$

The inverse of H can be computed by algorithm 1 which utilizes numeric estimation. Since attention mechanism can be regarded as a residual structure, same logic of Lemma 1 can be applied to design the invertible attention for our Shift-Elimination Module.

Next, we will introduce invertible Convolution. Since attention utilize Convolution to obtain feature map and response map, to make the whole network invertible, it is obligatory to the the Convolution invetible as well. We can represent the Convolution as $g(\mathbf{x}_{t-T:t}^{(n),d}) = W\mathbf{x}_{t-T:t}^{(n),d}$, where W is the weight matrix. We can compute the largest singular value $\sigma(W)$ and apply Lipschitz constant constraint by:

$$W = \begin{cases} \frac{cW}{\sigma(W)} & if \frac{cW}{\sigma(W)} < 1 \\ W & else \end{cases} \tag{7}$$

Since attention mechanism can be regarded as a matrix multiplication between a response map and a feature map, applying Lipschitz constraint can be equivalent to find the Lipschitz bounds of the multiplication. [25] proposed the theorem as:

Theorem 1 : Let $\mathcal{X}$ be a normed vector space and let $\mathbb{R}^{m\times n}$ represents the set of $m \times n$ vector space over the real number $\mathbb{R}$. Let $F : \mathcal{X} \rightarrow \mathbb{R}^{m\times n}$, and $R : \mathcal{X} \rightarrow \mathbb{R}^{m\times m}$. Define $A : \mathcal{X} \rightarrow \mathbb{R}^{m\times n}$ by $A(x) = R(x)F(x)$ where $x \in \mathcal{X}$. The following properties are further assumed:

1.F is is Lipschitz-continuous with L_1-Lipschitz constant c_F.

2.R is is Lipschitz-continuous with L_1-Lipschitz constant c_R.

3.$||R(x)||_1 \leq \mu_R$ for each $x \in \mathcal{X}$

4.$||F(x)||_1 \leq \mu_F$ for each $x \in \mathcal{X}$.

Then A has a L_1-Lipschitz constant $\mu_R c_F + c_R \mu_F$.

Due to the fact that the L_1 norm and L_2 norm are within constant bounds of each other as well as the sufficient and necessary condition of a function is L1 Lipschitz continuous is that it is also L2 Lipschitz continuous, this theorem can be extended to L_2 norm. We are not going to discuss the detailed proofs which can be seen in [25]. Another Lipshictz-constrained convolution can be added at the end of residual branch with the L_2 Lipschitz constant $\mathbb{L}(A) = k(\mu_R c_F + c_R \mu_F)$, where k is the constant bound. The attention block can be set as:

$$f(\mathbf{x}_{t-T:t}^{(n),d}) = \mathbf{x}_{t-T:t}^{(n),d} + W_A A(\mathbf{x}_{t-T:t}^{(n),d}) \tag{8}$$

where W_A is the weight. By further setting Lipschitz constant as $\frac{c}{k(\mu_R c_F + c_R \mu_F)}$, where $c \in (0,1)$, the residual branch satisfies: $\mathbb{L}(W_L A(\mathbf{x}_{t-T:t}^{(n),d}))\mathbb{L}(W_L \mathbf{x}_{t-T:t}^{(n),d})\mathbb{L}(A(\mathbf{x}_{t-T:t}^{(n),d})) = c$ As a result, the residual branch satisfies Lemma 1, the whole attention block is invertible and $f(\mathbf{x}_{t-T:t}^{(n),d})$ can be inversed by Algorithm 1.

Next, we will discuss the coupling layers of Spatial-Temporal Attention Flows which is built upon Normalizing Flows. Normalizing Flows are a set of mappings denoted as $f : \mathcal{X} \rightarrow \mathcal{Z}$ such that a simple distribution $p_{\mathcal{Z}}(\mathbf{z})$ on the space $\mathcal{Z} \in \mathbb{R}^N$ (Normally, $p_{\mathcal{Z}}(\mathbf{z})$ is defined as isotropic Gaussian distribution or uniformly distribution, but in this work, we do not use a pre-defined distribution) can be transformed from the densities $p_{\mathcal{X}}(\mathbf{x})$ on the input space $\mathcal{X} \in \mathbb{R}^N$. The density $p_{\mathcal{X}}(\mathbf{x})$ can be expressed as: $p_{\mathcal{X}}(\mathbf{x}) = p_{\mathcal{Z}}(\mathbf{z}) \left| det\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)\right|$ where$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ is the Jacobian matrix of f at $\mathbf{x}$.

Since mappings f consists of a sequence of bijections functions (i.e. $f = f_1 \circ f_2 \circ ... \circ f_n$), the inverse function $\mathbf{x} = f^{-1}(\mathbf{z})$ can be easily calculated. So the relationship between x and z can be further expressed as: $\mathbf{x} \overset{f_1}{\leftrightarrow} h_1 \overset{f_2}{\leftrightarrow} h_2 ... \overset{f_n}{\leftrightarrow} \mathbf{z}$ So the log-likelihood of $\mathbf{x}$ can be written as $log\ p_{\mathcal{X}}(\mathbf{x}) = log\ p_{\mathcal{Z}}(\mathbf{z}) + \sum_{i=1}^{n} log\ \left|det\left(\frac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-1}}\right)\right|$where we define $\mathbf{h}_0 := \mathbf{x}$ and $\mathbf{h}_n := \mathbf{z}$.

We can always transform any input densities into not only isotropic Gaussian distribution or uniform distribution but also any densities which are favorable for the training of models. As a result, unlike the normal training strategy of Normalizing Flows, i.e. calculating the log-likelihood of $p_{\mathcal{X}}(\mathbf{x})$, we train these two flows by minimizing the loss of the whole framework which we will discuss later. In addition, we also give up using the Jacobian matrix since we don't need to utilize the log-likelihood as the loss function.

C. Correlation Processing Module

After the shift effect is removed and the space is recovered to stationary space, we also aim to capture correlation. Since there are no non-stationary factors existing, the process of capturing correlation can be easier to pursue. Specifically, we intend to capture the correlation of the pre-defined graph as graph embedding as well as the correlation of POI as map embedding. Recent researches have demonstrated that Graph Convolutional Network (GCN) has a great advantage to model complicated graph patterns [32]. Thus, to capture the dependencies of pre-defined adjacency matrix, we utilize a GCN with Chebyshev polynomial expansion to model the complex spatial correlation. POI scattered in the map is also essential for the urban prediction task. To embed it, we divide cities into 10000 grids, each grid contains four categories of POI. We use a Convolutional layer to scan the most relevant grids. And ultimately, we include two learnable parameters to represent the weights of graph embedding and the map embedding for the feature fusion process.

Graph Embedding for Topological Correlation In this work, the network of the city(e.g., traffic network, air quality station network) is a graph structure, and we can view the features of each node(e.g., PM2.5 index) are the signal of the network. To utilize the entire topological properties of the urban network, we employ Graph Convolutional Network to directly process the signals. We generally represent a graph by its Laplacian matrix L which is defined as $L = D - \mathcal{A}$. Also we define its normalized form as $L = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where I_n is the identity matrix, $D \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix with $D_{ii} = \sum_{\forall j} W_{ij}$. $\Lambda \in \mathbb{R}^{N \times N}$ is the diagonal matrix of eigenvalues of L.

We denote the input as $\widetilde{\mathbf{z}}_t^{(1:N),d} \in \mathbb{R}^N$ and $\widetilde{\mathbf{z}}_t^{(1:N),d} \in \widetilde{\mathbf{Z}}_{t-T:t}^{(1:N)}$ which is the output of Shift Elimination Module. Then we refer to graph convolution operator as the notation $*_\mathcal{G}$ by using U as the graph Fourier basis:

$$\begin{aligned}\Theta_{*_\mathcal{G}}(\widetilde{\mathbf{z}}_t^{(1:N),d}) &= \Theta(L)\widetilde{\mathbf{z}}_t^{(1:N),d} \\ &= U\Theta(\Lambda)U^T\widetilde{\mathbf{z}}_t^{(1:N),d}\end{aligned} \tag{9}$$

where the kernel is Θ, $\Lambda \in \mathbb{R}^{N \times N}$ is the diagonal matrix of eigen- values of L, and $U \in \mathbb{R}^{N \times N}$ is the matrix of eigenvectors of the normalized graph Laplacian L.We use the 1^{st} order approximation of graph Laplacian [33] to simplified (we assumed the maximum value of λ is 2) and denote θ_0, θ_1 are learnable parameters of the kernel. For the stability, we replace θ_0 and θ_1 by $\theta = \theta_0 = -\theta_1$ and normalize W and D as $\widetilde{W} = W + I - n$ and $\widetilde{D} = \sum_j \widetilde{W}_{ij}$:

$$\begin{aligned}\Theta_{*_\mathcal{G}}(\widetilde{\mathbf{z}}_t^{(1:N),d}) &\approx \theta_0\widetilde{\mathbf{z}}_t^{(1:N),d} - \theta_1(D^{-\frac{1}{2}}WD^{-\frac{1}{2}})\widetilde{\mathbf{z}}_t^{(1:N),d} \\ &= \theta(I_n + D^{-\frac{1}{2}}WD^{-\frac{1}{2}})\widetilde{\mathbf{z}}_t^{(1:N),d} \\ &= \theta(\widetilde{D}^{-\frac{1}{2}}\widetilde{W}\widetilde{D}^{-\frac{1}{2}})\widetilde{\mathbf{z}}_t^{(1:N),d}\end{aligned} \tag{10}$$

By 1^{st} order approximation of graph Laplacian, the structure can be efficient.

Ultimately, we can derive the general form of Graph Convolutions:

$$\mathbf{h}_t^{(1:N),d_j} = \sum_{i=1}^{D} \Theta_{i,j}(L)\widetilde{\mathbf{z}}_t^{(1:N),d_i}, 1 \le j \le K \tag{11}$$

where $\widetilde{\mathbf{z}}_t^{(1:N),d_i} \in \mathbb{R}^N$ and $\mathbf{h}_t^{(1:N),d_j} \in \mathbb{R}^N$ are the input and output of the Graph Convolutions, D and F are the size of the input and output feature maps, and a and b are the positional index. L is the normalized graph Laplacian of the input adjacent matrix of the graph $\mathcal{A}$.

Map Embedding for Semantic Correlation For the $M * M$ grids of the cities, each grid contains C categories of POI information and each node is corresponded to one grid. Due to the remarkable ability to model Euclidean Structure Data, CNN has the advantage of modeling the POI data. Similar to the operation of pictures, we employ a simple Convolutional layer to scan the POI of F nearest grid of the node(including the grid where the node is in) which is the most relevant POI information and we can get a map embedding. So the POI information $\mathcal{P} \in \mathbb{R}^{C \times N}$ is transfer to $\mathcal{P}' \in \mathbb{R}^{C \times N \times F}$. We can formulate it as: $\check{\mathbf{H}}_{t-T:t}^{(1:N)} = conv2d(\mathcal{P}')$ where $\check{\mathbf{H}}_{t-T:t}^{(1:N)} \in \mathbb{R}^{T \times N \times F}$.

Multi-Feature Fusion To combine the information of POI and graph, we take the traffic speed as example. Traffic speed at some of the districts may rely more heavily on the nearby distracts rather than POI. An event such as the traffic jam can always slow down the traffic speed of all the nearby districts even the whole city. In contrast, POI may influence amounts of other districts more heavily compared to events of other districts. For instance, drivers need to slow down when they are approaching schools. We utilize learnable parameters W^G and W^C to reflect the weight of POI embedding and graph embedding for the features fusion. And we denote the output of GCN layers is $\mathbf{H}_{t-T:t}^{(1:N)}$. The fusion process can be formulated as:$\widetilde{\mathbf{H}}_{t-T:t}^{(1:N)} = W^G \odot \mathbf{H}_{t-T:t}^{(1:N)} + W^C \odot \check{\mathbf{H}}_{t-T:t}^{(1:N)}$, where $\odot$ is the Hadamard product, and $\widetilde{\mathbf{H}}_{t-T:t}^{(1:N)} \in \mathbb{R}^{T \times N \times F}$

Then, a linear layer NN is applied to change the shape of weighted embedding $\widetilde{\mathbf{H}}_{t-T:t}^{(1:N)}$: $\hat{\mathbf{H}}_{t-T:t}^{(1:N)} = NN(\widetilde{\mathbf{H}}_{t-T:t}^{(1:N)})$ where $\hat{\mathbf{H}}_{t-T:t}^{(1:N)} \in \mathbb{R}^{T \times N \times D}$ and $\widetilde{\mathbf{H}}_{t-T:t}^{(1:N)} \in \mathbb{R}^{T \times N \times F}$.

With the fusion feature, we can combine the information of both the graph and POI. This can significantly increase the ability of the framework to capture the spatial dependencies with more geographical information.

D. Forecasting Module

After eliminating the shift effect and capturing the correlation, the weighted embedding can be integrated into any spatial-temporal forecasting model which can be denoted as $\mathcal{F}_\theta$. The original prediction function $\hat{\mathbf{X}}_{t:t+\tau}^{(1:N)} = \mathcal{F}_\theta(\mathbf{X}_{t-T:t}^{(1:N)})$ can be replaced by the forecasting process which can be formulated as: $\hat{\mathbf{Y}}_{t:t+\tau}^{(1:N)} = \mathcal{F}_\theta(\hat{\mathbf{H}}_{t-T:t}^{(1:N)})$. The output here does not follow the original input space due to the shift-elimination process. So we need to further transfer the output $\hat{\mathbf{Y}}_{t:t+\tau}^{(1:N)}$ back to original input space $\mathcal{X}$, and then calculate the loss.

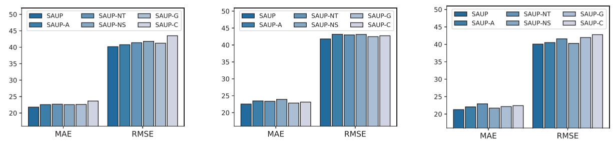
We will recover the shift information of the output $\hat{\mathbf{Y}}_{t:t+\tau}^{(1:N)}$ of the forecasting Module by the inverse transformation of Spatial-Temporal Attention Flows with the assumption that the input and output of this framework should belong the same space denoted as $\mathcal{X}$. We should notice that τ does not necessarily equal to T, in contrast, the coupling layers require the dimension of input and the dimensions of output should be equal. So we concatenate the output $\hat{\mathbf{Y}}_{t:t+\tau}^{(1:N)}$ and the vectors with zero value to get a new vector denote as $\hat{\mathbf{Y}}_{t:t+T}^{(1:N)}$. We use this vector as the input of the inverse transformation of the Shift Elimination Module and recover the shift information. Since the inverse function of the affine coupling layer proposed by RealNVP is easily to evaluate and the attention is invertible as well, we can easily obtain the result $\hat{\mathbf{X}}_{t:t+T}^{(1:N)}$. Ultimately, we split the output of the inverse process into two parts which are $\hat{\mathbf{X}}_{t:t+\tau}^{(1:N)}$ and $\hat{\mathbf{X}}_{t+\tau:t+T}^{(1:N)}$. $\hat{\mathbf{X}}_{t:t+\tau}^{(1:N)}$ is the final output of this framework.

TABLE I
OVERALL PERFORMANCE ON METR-LA DATASET.

METR-LA	Horizon 3			Horizon 6			Horizon 9			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
T-GCN	6.53	13.71	16.94%	7.48	15.07	20.25%	7.94	15.9	21.45%	8.27	16.44	21.97%
T-GCN + SAUP	**5.45**	**12.8**	**11.59%**	**5.95**	**13.85**	**12.17%**	**6.20**	**14.52**	**12.78%**	**6.66**	**15.26**	**14.25%**
ASTGCN	3.56	8.89	7.77%	4.45	11.42	9.30%	4.77	11.70	10.46%	5.73	13.01	13.01%
ASTGCN + SAUP	**3.37**	**8.55**	**7.49%**	**4.24**	**10.48**	**9.17%**	**4.59**	**11.09**	**10.26%**	**5.35**	**12.83**	**11.68%**
Transformer	5.43	12.63	10.59%	5.83	13.82	10.99%	5.98	13.81	11.53%	6.66	15.05	12.60%
Transformer + SAUP	**4.20**	**10.06**	**8.98%**	**5.11**	**12.02**	**10.93%**	**5.56**	**13.03**	**11.24%**	**6.26**	**14.31**	**12.66%**
Informer	5.64	13.02	12.10%	5.98	13.83	12.31%	6.46	14.38	13.42%	7.16	15.82	13.92%
Informer + SAUP	**4.22**	**10.46**	**8.92%**	**5.19**	**12.48**	**10.71%**	**5.67**	**13.45**	**11.47%**	**6.34**	**14.84**	**12.55%**
Dlinear	3.71	8.8	8.33%	4.71	10.77	10.45%	4.98	12.05	10.42%	5.85	13.06	12.97%
Dlinear + SAUP	**3.36**	**8.36**	**7.68%**	**4.02**	**10.16**	**9.04%**	**4.72**	**11.60**	**10.78%**	**5.42**	**12.76**	**11.90%**
N-BEATS	3.64	8.61	8.32%	4.48	10.51	9.96%	4.91	11.81	10.93%	5.72	13.28	12.65%
N-BEATS + SAUP	**3.38**	**8.44**	**7.73%**	**4.21**	**10.21**	**9.47%**	**4.65**	**11.55**	**10.44%**	**5.39**	**12.68**	**11.74%**

TABLE II
DATASET STATISTICS.

Datasets	# Samples	# Nodes	Sample Rate	Input Length	Output Length
METR-LA	34272	207	5 min	12	12
Beijing	33642	35	1 hour	12	12



(a) N-BEATS. (b) Transformer. (c) ASTGCN.

Fig. 3. Ablation study on the Beijing dataset.

E. *Optimization Strategy*

As mentioned in previous subsection, normal training strategy of Normalizing Flows is calculating the log-likelihood of $p_{\mathcal{X}}(\mathbf{x})$. However, in this work, we do not utilize Normalizing Flows to estimate density, so it is unnecessary to calculate the log-likelihood. Instead, we use Mean Absolute Error (MAE) as our loss function, and our objective is to minimize the overall loss $\mathcal{L}$ as follows:

$$\mathcal{L} = \frac{1}{\tau}\sum_{i=t}^{t+\tau}\left|\mathbf{X}_i^{(1:N)} - \hat{\mathbf{X}}_i^{(1:N)}\right| \tag{12}$$

IV. EXPERIMENTS

This section presents the empirical results of our proposed framework based on two real-world datasets. These experiments intend to answer the following questions: **Q1.** Does our framework (SAUP) improve the performance of backbones at a convincing level? **Q2.** Is each component of the model effective to SAUP? **Q3.** Is SAUP adequately robust? **Q4.** Is SAUP sensitive to the hyperparameters of different components? **Q5.** Can SAUP be trained efficiently?

A. *Experimental Setup*

In this subsection, we will give a brief summary of our datasets, and introduce the metrics and backbones we utilize in our experiments.

1) Data Description: We conduct experiments on two real-world datasets indicated by Table II: (1) **METR-LA** [3]: This traffic dataset includes statistics on traffic gathered from loop detectors on county highways in Los Angeles. We employ the speed of vehicles from Mar 1st, 2012 to Jun 30th, 2012 for the experiment. (2) **Beijing**: This air quality dataset contains several air quality indexes collected hourly in Beijing. We choose the PM2.5 index ranging from December 6th, 2013 to November 24 2017 for the experiment.

2) Evaluation Metrics: We evaluate the performance of the urban prediction task by three evaluation metrics. Let $\mathbf{X}_t^{(1:N)} \in \mathbb{R}^{N\times D}$ be the ground truth of all nodes at time step t, and $\hat{\mathbf{X}}_t^{(1:N)} \in \mathbb{R}^{N\times D}$ be the prediction result. The metrics can be defined as follows: Root Mean Square Error (RMSE) is given by $RMSE = \sqrt{\frac{1}{\tau}\sum_{i=t}^{t+\tau}\left(\mathbf{X}_i^{(1:N)} - \hat{\mathbf{X}}_i^{(1:N)}\right)^2}$; Mean Absolute Error (MAE) is given by $MAE = \frac{1}{\tau}\sum_{i=t}^{t+\tau}\left|\mathbf{X}_i^{(1:N)} - \hat{\mathbf{X}}_i^{(1:N)}\right|$; Mean Absolute Percentage Error (MAPE) is given by: $MAPE = \frac{1}{\tau}\sum_{i=t}^{t+\tau}\left|\frac{\mathbf{X}_i^{(1:N)} - \hat{\mathbf{X}}_i^{(1:N)}}{\mathbf{X}_i^{(1:N)}}\right|$.

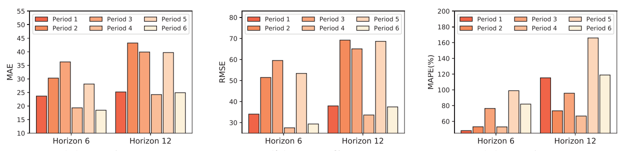
3) Backbones: We include the following widely used deep learning models as our backbones:
(1) Transformer [29] is able to capture the correlation among various sequential data by connecting the encoder and decoder jointly with the attention mechanism.
(2) Informer [34] is a time-efficient attention-based model that can model long sequential data.
(3) Dlinear [35] is a linear model that can model the trend and seasonality patterns of data.
(4) N-BEATS [36] is a simple network consisting of fully connected layers which can outperform in forecasting.
(5) ASTGCN [15] utilizes spatial and temporal attention for improving the ability to capture correlation among dynamic spatial-temporal series.
(6) T-GCN [14] can model spatial-temporal correlation with the graph convolutional network as well as the gated recurrent unit.

B. *Overall Performance*

The final outcome of the Shift-Aware Ubran Predication framework with diverse backbones on all two datasets is reported in Table I and Table III. In general, with the help

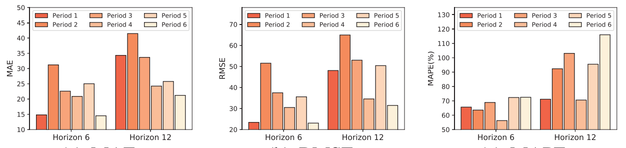
TABLE III
OVERALL PERFORMANCE ON BEIJING DATASET.

Beijing	Horizon 3			Horizon 6			Horizon 9			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
T-GCN	16.33	30.43	52.51%	19.99	36.89	68.67%	22.64	41.06	89.82%	24.87	44.73	99.53%
T-GCN + SAUP	**13.24**	**24.78**	**41.83%**	**17.48**	**32.67**	**58.97%**	**19.81**	**36.92**	**71.02%**	**21.88**	**40.28**	**79.77%**
ASTGCN	11.51	24.11	37.83%	15.80	31.69	56.85%	19.89	38.52	78.34%	23.11	43.07	94.47%
ASTGCN + SAUP	**11.24**	**22.53**	**37.42%**	**15.19**	**29.72**	**50.44%**	**19.43**	**36.31**	**71.92%**	**21.28**	**40.07**	**80.78%**
Transformer	13.29	25.65	39.74%	17.28	32.21	57.73%	20.06	38.63	75.62%	26.82	47.19	103.14%
Transformer + SAUP	**11.91**	**23.11**	**36.80%**	**16.40**	**31.37**	**55.52%**	**19.56**	**36.28**	**69.11%**	**22.55**	**41.79**	**80.28%**
Informer	13.61	26.96	42.75%	17.84	34.37	59.49%	20.30	37.75	79.18%	25.82	45.62	101.33%
Informer + SAUP	**11.86**	**22.71**	**36.20%**	**16.62**	**31.70**	**55.63%**	**19.26**	**35.76**	**79.18%**	**23.20**	**41.52**	**84.58%**
Dlinear	14.57	28.14	50.57%	18.88	36.11	67.66%	21.58	39.99	90.19%	25.83	46.19	99.71%
Dlinear + SAUP	**11.77**	**23.33**	**39.32%**	**16.44**	**31.76**	**57.22%**	**19.69**	**36.62**	**69.45%**	**23.22**	**41.51**	**87.05%**
N-BEATS	12.77	24.62	39.05%	16.83	34.00	56.41%	19.87	38.29	76.23%	22.50	42.16	90.89%
N-BEATS + SAUP	**11.84**	**23.49**	**39.26%**	**16.33**	**31.52**	**55.78%**	**19.65**	**36.26**	**69.80%**	**21.80**	**40.17**	**83.26%**



(a) MAE. (b) RMSE. (c) MAPE.

Fig. 4. Robustness Check for Transformer on the Beijing dataset evaluated.



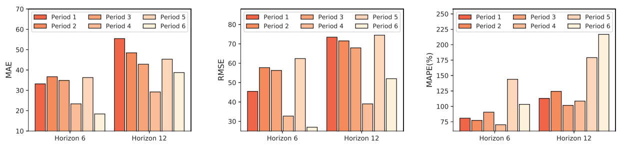
(a) MAE. (b) RMSE. (c) MAPE.

Fig. 5. Robustness Check for N-BEATS on the Beijing dataset evaluated.

of our proposed framework, all of the backbones have remarkable improvement (at least 5%) in terms of all three metrics on both datasets. Especially, with Horizon as 3 on METR-LA dataset, our proposed framework remarkably improves the performance (measured by MAE) of Informer by 25.18%(5.64⇒4.22) and ameliorates the performance (measured by MAE) of Transformer by 22.65%(5.43⇒4.2). In addition, the performance of ASTGCN with Horizon as 6 on the Beijing dataset is improved by 18.42% (18.62⇒15.19). This result demonstrates empirically that eliminating ubiquitous shift effect is essential for urban spatial-temporal prediction. Our proposed Shift-Aware Urban Prediction framework not only takes the shift effect into account and effectively eliminates both of spatial shift effect and temporal shift effect, but also captures the topological correlation and geographic correlation by combining the information of graph embedding and map embedding. However, our backbones can only capture the correlation among spatial-temporal series, thus decreasing the accuracy of their predictions.

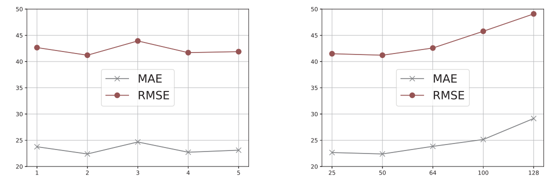
C. Ablation Study

To evaluate the effectiveness of each component of our proposed Shift-Aware Urban Prediction framework (SAUP), we conduct the ablation study on the Beijing dataset. We denote **SAUP-A** as the variant of SAUP without invertible attention. **SAUP-N** is another variant of SAUP without coupling layers. **SAUP-G** is the variant which removes GCN



(a) MAE. (b) RMSE. (c) MAPE.

Fig. 6. Robustness Check for ASTGCN on the Beijing dataset evaluated.



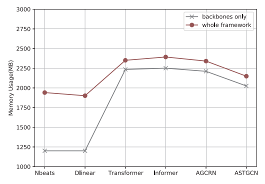
(a) Kernel Size of GCN layers. (b) Hidden Layers Size of Coupling Layers.

Fig. 7. Parameters Sensitivity study of Nbeats on the Beijing dataset.

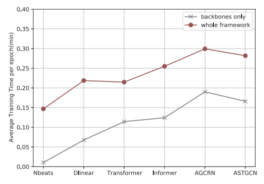
layers and **SAUP-C** is the variant which drops CNN layers. Figure 3 demonstrates that removing different components causes a degradation for diverse backbones, which indicates all components of our proposed framework are effective in eliminating shift effect as well as capturing correlation for pre-defined graph and POI.

D. Robustness Check

We also conduct the robustness check for our approach. By utilizing diverse subgroups of datasets, our framework is applied to examine the variance of performance. Specifically, The Beijing dataset is equally split into six time periods, and we employ four backbones with our framework to make a prediction for each time period on two horizons. Figure 6 shows the experiment results that the performance of our framework fluctuates in the different time periods. From our perspective, this phenomenon is related to the environmental policy of government. Stricter environmental policies were adopted in 2014, and gradually took effort in the following years. Hence, the first three subgroups of the dataset suffer from a serious temporal shift effect due to the instability of policies at the early stage. And by the fourth time period, the policy had become increasingly stable, allowing our framework to function effectively and steadily. It indicates the fact



(a) Memory Usage.



(b) Average Training Time per epoch.

Fig. 8. Training Efficiency study on the Beijing dataset where we record memory usage and training time.

that although our framework can eliminate the shift effect, unexpected incidents might also deteriorate the performance of the framework.

E. Parameters Sensitivity

In this subsection, we analyze the sensitivity of the hidden layer size of coupling layers and the kernel size of the GCN layer. To control the variable, We fix the hidden layer size as 50 when we adjust the kernel size and set the kernel size as 2 when we adjust the hidden layer size. We use N-BEATS as our backbone and conduct this study on the Beijing dataset. Figure 7 shows the performance of our proposed framework changes slightly with kernel size. In contrast, for the hidden layer size, the MAE and RMSE decrease when the hidden layer size changes from 25 to 50, but it gets larger when we increase the size of hidden layers. The potential reason for this trend is increasing the size of hidden layers in a certain level can improve the performance of framework but when size of hidden layers gets large enough, it provides additional training challenges to impede the convergence of model. So our framework is not sensitive to the kernel size of GCN layers but sensitive to the change of size of hidden layers.

F. Training Efficiency

Figure 8 indicates the memory usage and average training time for each epoch of the Beijing dataset. We can see that the maximum memory usage is less than 2400MB, and the maximum average training time for each epoch is less than 0.3min. It demonstrates that our framework is efficient to train for diverse backbones and easy to duplicate in other devices.

V. Related Work

Urban Prediction. For the purpose of providing urban construction suggestions, Urban prediction employs urban data to model complicated spatial-temporal patterns. There are several tasks of Urban prediction, for instance, traffic prediction [2], [12], [37], air quality prediction [38], vehicle charging demand [39], [40].

Previous works have demonstrated theoretically and empirically that modelling the correlation among spatial-temporal urban data can be an efficient way to reveal the hidden patterns among them. Some of them use GCN-based models to capture spatial-temporal information, such as [6], and some of them utilize statistical method, such as [41]. Their works contribute a lot to promoting the understanding of humans towards urban data. However, urban environment can be full of heterogeneous and dynamic information, so the shift effect is ubiquitous among urban data which will disturb the correlation-capturing process. In this work, we design a Shift Aware Urban Prediction (SAUP) framework to eliminate the shift effect. With our developed SAUP framework, spatial-temporal data can be converted to stationary space for the forecasting task, and then recover the shift information ultimately.

In addition, previous researches focus on modelling dependencies by either graph-based data [6] or grid-based data [27] but they seldom consider POI information and graph structure at the same time. In contrast, either semantic information or graph structure information plays an essential role in the urban prediction task. To tackle it, we utilize Correlation Processing Module containing Graph Convolutional Network (GCN) [33] and a Convolutional Neural Network (CNN) to fuse the features of POI information and graph structure. As a result, our forecasting module can better model the hidden correlation among urban spatial-temporal data.

Normalizing Flows. Since the proposal of NICE [23] and Real NVP [24], Normalizing Flows have never failed to fascinate researchers due to their remarkable ability to transform distributions. Previously, researchers utilize Normalizing Flows for the purpose of density estimation [22]and variational inference [42]. Real NVP [24] proposes an affine coupling layer to improve the estimation ability of flows. [21] designs flows consisting of actnorm, invertible 1×1 convolution and affine coupling layer. Experiments empirically indicate that Normalizing Flows can preform well in these tasks.

However, preceding studies do not employ Normalizing Flows to obliterate non-stationary factors . In this work, we construct Spatial Temporal Attention Flow (STAF) as tool to eliminate the shift effect. Two coupling layers are used to transfer the original distribution of input which contains shift to a hidden distribution which is without shift and recover the shift information after the Forecasting Module performs the prediction task. With the help of Normalizing Flows, the framework can capture the spatial-temporal correlation more precisely in a stationary space.

VI. Conclusion

In this work, we propose Shift-Aware Urban Prediction framework to solve the problem of intrinsic shift among spatial-temporal data. First, we construct a Shift-Elimination Module consisting of invertible attention and coupling layers to eliminate spatial and temporal shift effects. Then we proposed the Correlation Processing Module to capture correlation among urban data. Specifically, we utilize a Convolutional layer to embed the POI information of the map and use a GCN layer to capture the hidden spatial pattern behind the adjacency matrix. We combine these two pieces of information together to achieve the goal of better feature fusion. After, we use a model-agnostic Forecasting Module for prediction. At last, we utilize the inverse transformation of the Shift-Elimination Module to recover the shift information. Extensive experiments on real-world datasets show that with the proposed Shift-Aware Urban Prediction framework, the performance of the backbone models can be improved.

Acknowledgment

This work is supported by the Science and Technology Development Fund (FDCT), Macau SAR (File no. SKL-IOTSC-2021-2023), and the Start-up Research Grant of University of Macau (File no. SRG2021-00017-IOTSC).

References

[1] S. H. A. Jarah, B. Zhou, R. J. Abdullah, Y. Lu, and W. Yu, "Urbanization and urban sprawl issues in city structure: A case of the sulaymaniah iraqi kurdistan region," *Sustainability*, vol. 11, no. 2, p. 485, 2019.

[2] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.

[3] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[4] W. Fan, S. Zheng, X. Yi, W. Cao, Y. Fu, J. Bian, and T.-Y. Liu, "DEPTS: Deep expansion learning for periodic time series forecasting," in *International Conference on Learning Representations*, 2022.

[5] A. Camero and E. Alba, "Smart city and information technology: A review," *cities*, vol. 93, pp. 84–94, 2019.

[6] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 804–17 815, 2020.

[7] H. Z. Moayedi and M. Masnadi-Shirazi, "Arima model for network traffic prediction and anomaly detection," in *2008 international symposium on information technology*, vol. 4. IEEE, 2008, pp. 1–6.

[8] Z. Lu, C. Zhou, J. Wu, H. Jiang, and S. Cui, "Integrating granger causality and vector auto-regression for traffic prediction of large-scale wlans," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 10, no. 1, pp. 136–151, 2016.

[9] W. Liu, K. Seto, and Z. Sun, "Urbanization prediction with an art-mmap neural network based spatio-temporal data mining method," in *Proceedings of the ISPRS Joint Conference 3rd International Symposium Remote Sensing and Data Fusion Over Urban Areas and 5th International Symposium Remote Sensing of Urban Areas, Tempe*, 2005.

[10] L. Bai, L. Yao, S. Kanhere, X. Wang, Q. Sheng *et al.*, "Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," *arXiv preprint arXiv:1905.10069*, 2019.

[11] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[12] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[13] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *arXiv preprint arXiv:1703.04691*, 2017.

[14] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[15] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[16] L. Bai, L. Yao, S. S. Kanhere, Z. Yang, J. Chu, and X. Wang, "Passenger demand forecasting with multi-task convolutional recurrent neural networks," in *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part II 23*. Springer, 2019, pp. 29–42.

[17] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[18] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *AAAI*. AAAI Press, 2023.

[19] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*, 2021.

[20] W. Fan, P. Wang, D. Wang, D. Wang, Y. Zhou, and Y. Fu, "Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7522–7529.

[21] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.

[22] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," *Advances in neural information processing systems*, vol. 30, 2017.

[23] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[24] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[25] J. Zha, Y. Zhong, J. Zhang, R. Hartley, and L. Zheng, "Invertible attention," *arXiv preprint arXiv:2106.09003*, 2021.

[26] C. Cangea, P. Veličković, N. Jovanović, T. Kipf, and P. Liò, "Towards sparse hierarchical graph classifiers," *arXiv preprint arXiv:1811.01287*, 2018.

[27] P. Zhao, A. Luo, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, and X. Zhou, "Where to go next: A spatio-temporal gated network for next poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2512–2524, 2020.

[28] H. Lin, Z. Gao, Y. Xu, L. Wu, L. Li, and S. Z. Li, "Conditional local convolution for spatio-temporal meteorological forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 7470–7478.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[31] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J.-H. Jacobsen, "Invertible residual networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 573–582.

[32] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," *arXiv preprint arXiv:2002.05287*, 2020.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[34] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[35] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" 2023.

[36] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[37] J. Ming, L. Zhang, W. Fan, W. Zhang, Y. Mei, W. Ling, and H. Xiong, "Multi-graph convolutional recurrent network for fine-grained lane-level traffic flow imputation," in *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2022, pp. 348–357.

[38] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, "Deep distributed fusion network for air quality prediction," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 965–973.

[39] F. B. Hüttel, I. Peled, F. Rodrigues, and F. C. Pereira, "Deep spatio-temporal forecasting of electrical vehicle charging demand," *arXiv preprint arXiv:2106.10940*, 2021.

[40] W. Zhang, H. Liu, J. Han, Y. Ge, and H. Xiong, "Multi-agent graph convolutional reinforcement learning for dynamic electric vehicle charging pricing," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 2471–2481.

[41] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, "Forecasting traffic congestion using arima modeling," in *2019 15th international wireless communications & mobile computing conference (IWCMC)*. IEEE, 2019, pp. 1227–1232.

[42] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," *Advances in neural information processing systems*, vol. 29, 2016.