Generative-Free Urban Flow Imputation

Senzhang Wang*
Central South University
Changsha, China
szwang@csu.edu.cn

Jiyue Li*
Nanjing University of Aeronautics and Astronautics
Nanjing, China
lijiyue@nuaa.edu.cn

Hao Miao
Aalborg University
Aalborg, Denmark
haom@cs.aau.dk

Junbo Zhang
JD Intelligent Cities Research
JD iCity, JD Technology
Beijing, China
msjunbozhang@outlook.com

Junxing Zhu
National University of Defense Technology
Changsha, China
zhujunxing@nudt.edu.cn

Jianxin Wang
Central South University
Changsha, China
jxwang@mail.csu.edu.cn

ABSTRACT

Urban flow imputation, which aims to infer the missing flows of some locations based on the available flows of surrounding areas, is critically important to various smart city related applications such as urban planning and public safety. Although many methods are proposed to impute time series data, they may not be feasible to be directly applied on urban flow data due to the following reasons. First, urban flows have the complex spatial and temporal correlations which are much harder to be captured compared with time series data. Second, the urban flow data can be random missing (i.e., missing randomly in terms of times and locations) or block missing (i.e., missing for all locations in a particular time slot). Thus it is difficult for existing methods to work well on both scenarios. In this paper, we for the first time study the urban flow imputation problem and propose a generative-free Attention-based Spatial-Temporal Combine and Mix Completion Network model (AST-CMCN for short) to effectively address it. Specifically, AST-CMCN consists of a Spatial and Temporal Completion Network (SATCNet for short) and a Spatial-Temporal Mix Completion Network (STMCNet for short). SATCNet is composed of stacked GRUAtt modules to capture the geographical and temporal correlations of the urban flows, separately. STMCNet is designed to capture the complex spatial-temporal associations jointly between historical urban flows and current data. A Message Passing module is also proposed to capture new spatial-temporal patterns that never appear in the historical data. Extensive experiments on two large real-world datasets validate the effectiveness and efficiency of our method compared with the state-of-the-art baselines.

*Both authors contributed equally to this research.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557334

CCS CONCEPTS

• **Information systems → Data mining**; **Spatial-temporal systems**; • **Networks → Location based services**.

KEYWORDS

Spatial-temporal, Urban flow imputation, Attention mechanism

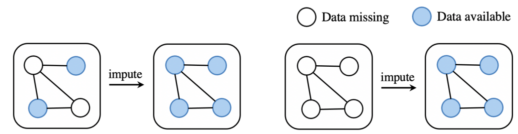
ACM Reference Format:

Senzhang Wang, Jiyue Li, Hao Miao, Junbo Zhang, Junxing Zhu, and Jianxin Wang. 2022. Generative-Free Urban Flow Imputation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557334

1 INTRODUCTION

Urban flows (e.g., taxi flows, bike flows and human trajectories), which depict the detailed human mobility patterns in urban areas, are critically important to many smart city related applications such as urban planning, city renewal and traffic management [1, 2]. To obtain the urban flow observations, a large number of sensors such as cameras are required to be deployed in different locations all over the city. However, in real scenarios, the complete urban flow data is usually unavailable with partial missing data due to various reasons including sensor errors (e.g. power outages) or communication errors [3]. For example, urban flows of all sensors are transmitted from edge nodes (sensors) to data storage center regularly (like every 5 minutes). Data might be lost during transmission due to communication errors. Generally, the missing data issue can be categorized into two types as illustrated in Figure 1: *a)* ***Random missing****: the urban flow data is missing randomly at arbitrary sensors or time slots as shown in Figure 1(a). b)* ***Block missing****: the readings of all sensors are all missing in one or several timestamp(s) simultaneously as shown in Figure 1(b).* Therefore, how to effectively impute the incomplete urban flow observations has become a critical research issue and attracted rising research attention recently.

Traditionally, statistic methods are widely used for data imputation, such as K-Nearest Neighbors (KNN) [4], Matrix Factorization (MF) [5] and Multiple Imputation using Chained Equations (MICE) [6]. However, these methods ignore the complex spatial-temporal properties of urban flow data, leading to suboptimal performance. Recently, motivated by the great success of generative models [7, 8],



(a) Random missing data imputation.

(b) Block missing data imputation.

Figure 1: Two missing data scenarios. White circles denote the data are missing and blue ones denote the data are available.

researchers tried to impute missing data using generative-based models. GAIN [9] for the first time employed generative adversarial network [7] for data imputation. Qi et al. [10] proposed NAM, a recurrent conditional variational autoencoder method for trajectory imputation. MCflow [11] is a deep generative-based model for data imputation, which leveraged normalizing flow generative models and Monte Carlo sampling. Zhang et al. [12] proposed SA-GAIN based on self-attention generative model for traffic flow imputation. However, the major limitation of the generative models is that they are relatively hard to train and converge in training. To address this issue, MVTS [13] began to employ transformer-based method for time series data imputation. MVTS utilized attention mechanism to capture the temporal correlations of time series data. To further improve the performance of MVTS, SAITS [14] designed a diagonally-masked self-attention (DMSA) block and a joint-optimization approach to capture both the temporal dependencies and feature correlations between time steps. However, the above methods are mostly designed for time series data imputation, which may not be feasible to be directly applied on urban flow imputation due to the complex spatial-temporal patterns of the urban flows.

Compared with the previous time series data or trajectory data imputation, the studied urban flow imputation is much more difficult due to the following challenges.

- **The backbone network for urban flow imputation.** For the data imputation problem, the existing state-of-the-art models generally take the generative model as the backbone network. However, most of the generative models, such as Generative Adversarial Network (GAN) [7] and Variational Auto Encoder (VAE) [8], need to be well-designed and are relatively hard to train. Directly applying the generative model on urban flow imputation may be infeasible considering the complex spatial-temporal correlations of the data. Thus, how to design a generative-free model as the backbone network is challenging.
- **The complex spatial-temporal patterns of the urban flow data.** Existing works on time series data imputation only need to consider the temporal associations. Considering the complex spatial-temporal correlations of the urban flow data, learning the spatial and temporal correlations separately, and then fusing them together may not work well for the missing data scenario. Moreover, it is possible that some new spatial-temporal patterns of urban flows may have never appeared before and thus they cannot be captured from historical urban flow observations.
- **The two types of the missing data.** As shown in Figure 1(b), imputing the data for the block missing scenario is much more challenging than the random missing scenario as we cannot use the data in the neighbor locations to help infer the missing data based on the spatial correlations. Thus, the presence of both random and block missing will increase the difficulty of the urban flow data imputation. How to jointly impute the two types of missing data is challenging.

To tackle the aforementioned challenges, we present an Attention-based Spatial-Temporal Combine and Mix Completion Network model named AST-CMCN for generative-free urban flow data imputation. Specifically, we first propose a Spatial and Temporal Completion Network (SATCNet) to impute urban flow observations by capturing the spatial and temporal correlations among regions separately. SATCNet includes multiple layers of GRUAtt modules for capturing the temporal information, which is especially helpful to resolve the block missing data problem. The designed Gate module inner GRUAtt aims to acquire the spatial associations among regions to impute random missing data. Next, we design a Spatial-Temporal Mix Completion Network (STMCNet), which contains a Message Passing module to jointly capture the complex spatial-temporal patterns that cannot be easily captured by SATCNet. The Message Passing module is designed to capture the spatial-temporal patterns between historical urban flows and current data. New patterns that have not appeared before are captured by the designed super node structure. We summarize our contributions as follows.

- We for the first time study the urban flow imputation problem for both random missing and block missing scenarios and propose a generative-free model AST-CMCN to effectively address it.
- A Spatial and Temporal Completion Network SATCNet is proposed to impute urban flows by capturing spatial and temporal dependencies separately. A Spatial-Temporal Mix Completion Network STMCNet is also designed to capture the spatial-temporal associations of urban flows jointly.
- Extensive experiments are conducted on two large real-world datasets. Experimental results demonstrate the superiority of our method by comparison with the existing state-of-the-art approaches in terms of both random and block missing urban flow data imputation.

The remainder of the paper is organized as follows. We will first briefly review related work in Section 2. Then, notations and problem definition will be introduced in Section 3. Next, we will introduce the proposed model in detail in Section 4, followed by experimental results in Section 5. Finally, we will conclude the paper in Section 6.

2 RELATED WORK

2.1 Urban Flow Data Prediction

Recently, urban flow data prediction [15–17] has attracted rising research interest in the field of urban computing. Traditionally, statistics-based time series models such as ARIMA and Regression are used to address this problem. For example, [18] used ARIMA model to predict short-term urban flows. However, conventional methods usually are not effective to capture the complex spatial and

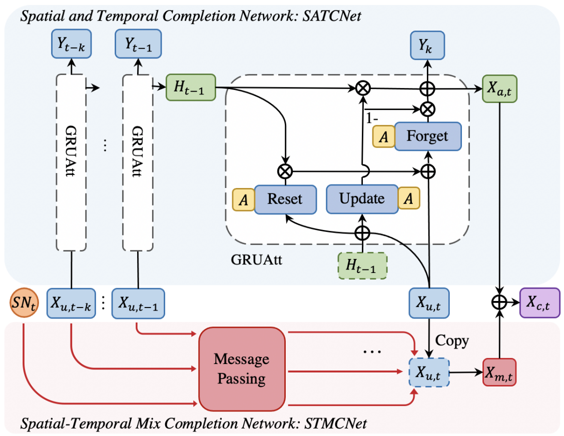


Figure 2: Framework of the proposed AST-CMCN model.

temporal correlations of urban flow data. Recently, various deep learning methods are proposed, and achieve significant prediction performance gains. ConvLSTM [19] employed both spatial and temporal information for the spatial-temporal data prediction based on LSTM [20]. In order to capture periodic properties, Zhang et al. [21] designed an end-to-end structure named ST-ResNet which contained a residual neural network and residual convolutional units. For learning historical information and making the prediction step by step, Pan et al. [22] designed an encoder-decoder architecture ST-MetaNet+ which combined meta-learning with deep learning. MT-ASTN [23] adopted a shared-private framework which contained private spatial-temporal encoders, a shared spatial-temporal encoder, and decoders to learn the task-specific features and shared features. Yin et al. [24] proposed MASTGN, including an internal attention mechanism and a dynamic neighborhood-based attention mechanism, which were used to capture the interactions among multiple time series and to model the complex spatial correlations. Different from the urban flow data prediction, the problem studied in this paper aims to impute rather than predict urban flows.

2.2 Time Series Data Imputation

Early time series data imputation methods are mostly statistics based, such as K-Nearest Neighbors (KNN) [4], Matrix Factorization (MF) [5] and Multiple Imputation using Chained Equations [6]. However, statistics based methods are less promising as they are not effective to capture the spatial and temporal properties of urban flow data. Recently, motivated by the great success of generative models [7, 8], researchers tried to impute missing data using generative-based model, since the data imputation is similar to data generative process. GAIN [9] for the first time employed generative adversarial network (GAN) [7] for data imputation. Luo et al. [25] proposed GANFilling model which was also based on GAN for data imputation. Mattei and Frellsen [26] proposed MIWAE for data imputation which was based on the importance-weighted autoencoder (IWAE) [27]. MIWAE maximized a potentially tight lower bound of the log-likelihood of the observed data without additional computational. NAM [10] designed a recurrent conditional variational autoencoder as a demonstrator and a non-autoregressive transformation model as a learner for trajectory imputation. MCflow [11] is a deep generative-based model, which leveraged normalizing flow generative models and Monte Carlo sampling for incomplete data. Zhang et al. [12] proposed a model for traffic flow imputation, which combined a self-attention mechanism, an auto-encoder and a generative adversarial network. However, the deep learning based generative models need to be well-designed and are hard to train.

Recently some works began to employ generative-free methods for data imputation. MVTS [13] was designed to capture temporal relationship of time series data under the framework of transformer. SAITS [14] used a diagonally-masked self-attention (DMSA) block and a joint-optimization approach to capture both the temporal dependencies and feature correlations between time steps. Nevertheless, these models are mostly designed for time series imputation, which cannot capture the complex spatial-temporal patterns of urban flow data.

3 NOTATIONS AND PROBLEM DEFINITION

We will first give some notations to help us state the studied problem. Then a formal problem definition will be given.

Definition 1. ***Urban flow graph.*** *The urban flow graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, ..., v_N\}$ is a set of N sensor nodes deployed in the city and $\mathcal{E}$ is a set of edges connecting these nodes. These connections between nodes can also be described by a symmetric adjacency $A \in \mathbb{R}^{N\times N}$, with the element $A_{i,j}$ representing the strength of the relation between nodes v_i and v_j. The relation strength is usually measured by the geographical proximity of the two sensors. Note that $A_{i,j} = 0$ if there is no close relation between the two nodes in geography. In addition, we denote the adjacency between node i at time $t-1$ and node j at time t as A_{i_{t-1},j_t}.*

Definition 2. ***Complete and incomplete urban flows.*** *We denote $X \in \mathbb{R}^{N\times D}$ as urban flow observations on urban flow graph $\mathcal{G}$. We denote incomplete and complete urban flows at time t as $X_{u,t} \in \mathbb{R}^{N\times D}$ and $X_{c,t} \in \mathbb{R}^{N\times D}$ respectively, where N is the number of nodes and D is dimension. The historical urban flows can be represented as a sequence $\mathcal{X}_u = (X_{u,t-k}, ..., X_{u,t})$ on the urban flow network. To conduct the urban flow data imputation over the regions where the data are unavailable, we also define the mask matrix of incomplete urban flows as follows:*

$$M_t(n) = \begin{cases} 0, & if\ X_{u,t}(n)\ is\ missing \\ 1, & otherwise \end{cases} \tag{1}$$

where n denotes the sensor node, $M_t(\cdot)$ is a mask function, which marks the nodes without urban flow observations as 0 and the nodes with data as 1.

The complex spatial-temporal patterns are captured from the historical urban flow observations first and then help impute the missing data. However, some new spatial-temporal patterns that exist outside the urban flow graph $\mathcal{G}$ cannot be captured from the historical data. Thus, we design a *super node* outside the graph to represent new urban flow data.

Definition 3. ***Super Node.*** *We denote $SN_t \in \mathbb{R}^{1\times D}$ as the super node at time t which represents the unseen urban flows on urban flow graph $\mathcal{G}$. SN_t is used to capture the new spatial-temporal patterns of urban flows.*

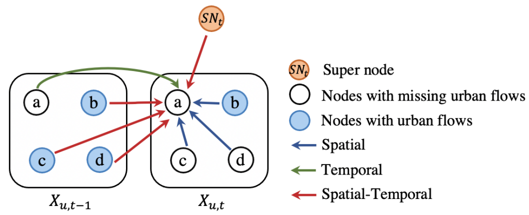


Figure 3: Correlations between nodes in two urban flow graphs.

The studied problem is formally defined as follows.

PROBLEM 1. *Given the urban flow graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the historical urban flow data $\mathcal{X}_u = (X_{u,t-k}, ..., X_{u,t})$ and a super node SN_t, we aim to build a model f to impute the incomplete urban flows $X_{u,t}$ and obtain a complete urban flow matrix $X_{c,t}$.*

In the following, we name the incomplete urban flows $X_{u,t}$ as the target urban flow data for convenience.

4 METHODOLOGY

Figure 2 shows the framework of the proposed AST-CMCN, which contains a Spatial and Temporal Completion Network (SATCNet) and a Spatial-Temporal Mix Completion Network (STMCNet). In SATCNet, as shown in the top of Figure 2, historical incomplete urban flows are input into the network. Then, multiple layers of GRUAtt are stacked to capture both temporal and spatial dependencies, separately. For the stacked GRUAtt modules, GRU is a backbone network to capture temporal correlations between the historical and target urban flows, which helps impute the block missing data. In the inner part of the GRUAtt module, *Reset, Update* and *Forget* Gate are designed with the same Gate module based on attention mechanism to obtain the geographical relationships among regions of incomplete urban flows. The Gate module completes urban flows for the random missing scenario. We will elaborate this part in Section 4.1.

Next, as shown in the bottom of Figure 2, historical urban flows are input into STMCNet similar to SATCNet. In order to acquire the spatial-temporal patterns that never appear in the historical data, a super node is added into the network as well. Here, we treat the super node as urban flow data outside the area under study and we cannot obtain it from the historical urban flows. Then, STMCNet designs a Message Passing module to capture the complex spatial-temporal patterns between the historical and target incomplete urban flows, and new patterns that never appear. In Message Passing module, we elaborate Transformer-based ODTrans module to capture the complex spatial-temporal correlations between regions in different time slots and regions. In this process, we impute urban flow data for both random and block missing scenarios. The STMCNet part will be introduced in detail in Section 4.2. SATCNet and STMCNet are jointly trained. Next, we will introduce the model in detail.

4.1 Spatial and Temporal Completion Network

Figure 3 shows the spatial-temporal correlations among the nodes. Green lines with arrow denote the temporal correlations of the same node in different time slots. Blue lines with arrow represent the spatial correlations of nodes in a urban flow graph based on geographical proximity. Red lines with arrow are the spatial-temporal correlations between nodes. As we can see node a at time slot t is correlated with nodes b, c, d at time slot $t-1$. For example, taxi flows on node d at time slot $t-1$ is correlated to taxi flows on node a at time slot t as they move from node d at $t-1$ to node a at t. In order to capture both the spatial and temporal correlations for urban flow data separately, we design a Spatial and Temporal Completion Network (SATCNet). The model that jointly captures the spatial-temporal correlations will be introduced in detail in Section 4.2.

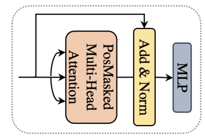


Figure 4: Gate module for the *Reset, Update* and *Forget* gates.

Temporal Completion. As shown in the top part of Figure 2, the sequence $\mathcal{X}_u = (X_{u,t-k}, ..., X_{u,t})$ are input into network. Motivated by the classical model GRU [28] which learns temporal correlations, we design stacked GRUAtt modules to obtain both the temporal and spatial dependencies. The designed module for capturing spatial correlations will be introduced in the next paragraph. By GRUAtt module, features extracted from historical urban flows are passed into hidden states and then sent into target urban flows. In the target urban flow data, incomplete data values are filled and adjusted by the hidden states. The GRUAtt module is beneficial to the block missing scenario, since we can complete the missing data of all regions at one timestamp by features extracted from the historical urban flows. Inside the GRUAtt module, we design a *Reset*, a *Update* and a *Forget* gate to effectively combine the target urban flows with previous hidden states (features of previous urban flows), choose the most useful information and forget insignificant messages. After that, we encode temporal correlations into the hidden states H_{t-1}. Note that the number of GRUAtt modules is equal to the number of input urban flow data. Formally, the GRUAtt module is formulated as follows:

$$r_t = \sigma(Gate([X_{u,t}||H_{t-1}], A)), \tag{2}$$

$$u_t = \sigma(Gate([X_{u,t}||H_{t-1}], A)), \tag{3}$$

$$c_t = \tanh(Gate([X_{u,t}||r_t \odot H_{t-1}], A)), \tag{4}$$

$$H_t = u_t \odot H_{t-1} + (1 - u_t) \odot c_t, \tag{5}$$

where H_{t-1} denotes the hidden state of previous urban flows, A denotes static adjacency matrix, $[\cdot||\cdot]$ means concatenation process, $\sigma(\cdot)$ is sigmoid function and $\tanh(\cdot)$ is tangent function. Note that when $t = 1$, H_0 is initialized with random values. $Gate(\cdot, \cdot)$ represents message passing process in geography and will be introduced later.

Spatial Completion. According to *Tobler's First Law of Geography* [29], *"All things are related, but nearby things are more related than distant things"*, it is intuitive that urban flow observations are influenced by regions in the geographical surroundings. Therefore, to obtain the spatial dependencies of urban flows in different sensors and complete urban flow data, we design a *Reset*, a *Update* and a *Forget* gate with the same Gate module, which is shown in Figure

4. The Gate module is implemented by the attention mechanism [30]. With the attention-based module, we can learn the arbitrary spatial dependencies between each pair of nodes in the urban flow graph. However, in the real scenario, a number of nodes are not reachable and connected directly. For example, nodes a and b are disconnected since there is a river between them or they are hundred kilometers away. Thus, we ought to employ the geographical adjacency matrix A to model all the spatial associations in urban flow graph. Then, we calculate the similarity of each connected pair of sensor nodes. With the similarity values, we fill target nodes with reachable nodes nearby by weighted summation. In this way, we capture the spatial correlations of urban flow data for imputing the random missing data. We formulate designed attention mechanism as:

$$Att(Q,K,V,A) = Softmax(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}} \otimes A)V, \tag{6}$$

$$head_i = Att(xW_i^Q, xW_i^K, xW_i^V, A), \tag{7}$$

$$MHAtt(x,A) = Concat(head_1, ..., head_h)W^H, \tag{8}$$

where Q, K, V are all the same input urban flows, h is the number of heads, $W_i^Q \in \mathbb{R}^{d_{model} \times d_q}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W^H \in \mathbb{R}^{hd_v \times d_{model}}$ and A denotes the adjacency matrix. Then the Gate module for the *Reset*, *Update* and *Forget* gate is as follows:

$$Gate(X,A) = MLP(MHAtt(X,A)), \tag{9}$$

where $MLP(\cdot)$ is a non-linear function. To enhance the ability of the Gate module, the mask ought to be applied inside the attention. As formulated in Eq. (10), the diagonal entries of the attention map are set as $-\infty$ to avoid unnecessary calculation.

$$Mask(a)(i,j) = \begin{cases} -\infty & i = j \\ a(i,j) & i \neq j \end{cases}, \tag{10}$$

where a is the learned attention map, i and j are the positions of the attention map. With Eq. (10), we replace Eq. (6) with Eq. (11) inside the Gate module as follows:

$$PosMaskedAtt(Q,K,V,A) = Softmax(Mask(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}) \otimes A)V, \tag{11}$$

Through obtaining the spatial relationship by the Gate module and then passing temporal messages from the historical urban flows to target ones, the missing urban flows are imputed. The final output of SATCNet is $X_{a,t}$. In SATCNet, we consider the missing types of urban flows. Obviously, the random missing urban flow data can be easily imputed by the designed Gate module. Spatial associations inside the Gate module might not help block missing problems, but the messages sent from the historical urban flows by stacked GRUAtt modules can help to solve the blocking missing problem.

4.2 Spatial-Temporal Mix Completion Network

Section 4.1 introduces how to capture the spatial associations and temporal correlations between nodes. However, the two types of associations are separately captured by SATCNet. Next we introduce a Spatial-Temporal Mix Completion Network (STMCNet) to capture the complex spatial-temporal correlations jointly.

In the bottom of Figure 2, the super node SN_t and the historical urban flows $X_u = (X_{u,t-k}, ..., X_{u,t})$ are input into the network. To capture the complex spatial-temporal relationships between nodes and then impute missing urban flows, we design a Message Passing module as shown in Figure 5. For the missing urban flow data, we wish to seek the similar spatial-temporal patterns from the historical urban flow data. Then in the Message Passing module, we impute the target urban flows by the corresponding urban flows which have similar spatial-temporal patterns. Besides, we also consider whether there are new spatial-temporal patterns which have never appeared in the historical urban flow observations. Thus, we design a super node SN_t as an unseen and new urban flow data and also send it into the Message Passing module. We treat the super node as a sensor node outside the whole urban flow graph $\mathcal{G}$, capturing correlations between nodes inside and outside the graph. As a result, the Message Passing module aims to capture the complex spatial-temporal patterns between the target urban flows $X_{u,t}$ and all the other input data (we call them auxiliary urban flows) and then imputes the target urban flow observations. With all the imputation results, we make a weighted summation. The Message Passing module is formulated as follows:

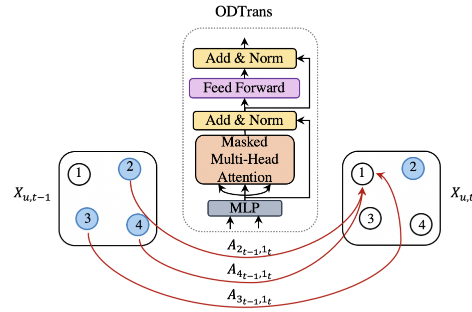


Figure 5: The Message Passing module.

$$X_t = Message(SN_t, X_{u,t_0}, ..., X_{u,t}), \tag{12}$$

$$\begin{aligned} Message(\dots) = &\sum_{k=t_0}^{t-1} W_k \cdot ODTrans(X_{u,k}, X_{u,t}) \\ &+ W_{k+1} \cdot ODTrans(SN_t, X_{u,t}), \end{aligned} \tag{13}$$

where W_k is the learnable parameter. We initialize the super node with random values. The auxiliary urban flow data might have similar patterns with the target urban flows, thus we design an attention-based mechanism ODTrans module shown in Figure 5. The ODTrans module aims to capture the complex spatial-temporal patterns and then impute target urban flows $X_{u,t}$. It is reasonable that all nodes in target urban flows ought to have spatial-temporal similarity values with nodes in the historical period (all nodes in $X_{u,t-k}, ..., X_{u,t-1}$). Thus, we use ODTrans to generate these similarity values. With the corresponding similarity values, we make a weighted summation and then impute the target urban flow data. For example, in Figure 5, we employ ODTrans module to capture the complex spatial-temporal similarity values between nodes 2, 3, 4 at time slot $t-1$ and node 1 at time slot t. Then we obtain the corresponding similarity weights $A_{2_{t-1},1_t}, A_{3_{t-1},1_t}, A_{4_{t-1},1_t}$. We make a weighted summation of urban flows of nodes 2, 3, 4 at time slot $t-1$ and similarity weights $A_{2_{t-1},1_t}, A_{3_{t-1},1_t}, A_{4_{t-1},1_t}$, to impute urban flows of node 1 at time slot t. Imputing the block missing flows cannot rely on geographical information. Thus, we impute missing urban flows by means of the complex spatial-temporal patterns regardless of region and time, which works for both random and block missing scenarios. ODTrans module can be conducted as follows:

$$ODTrans(ax, tg) = FFN(MaskedMHA(MLP(ax), MLP(tg))), \quad (14)$$

$$FFN(x) = ReLU(W_2(W_1 x + b_1) + b_2), \quad (15)$$

where $MLP(\cdot)$ is a non-linear function, W_1, W_2 are learnable parameters, ax is the auxiliary urban flow and tg is the target urban flow. The process of learning similarity weights is achieved by Masked MultiHeadAttention module (MaskedMHA) as follows:

$$MaskedMHA(ax, tg) = Concat(head_1, ..., head_h)W_H, \quad (16)$$

$$head_h = MaskedAtt(ax, tg, tg), \quad (17)$$

$$MaskedAtt(Q, K, V) = Softmax(Mask(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}))V, \quad (18)$$

where W_H is the learnable parameter. The final output of STMCNet is $X_{m,t}$. In STMCnet, the imputation process neither simply rely on geographical correlations nor single temporal correlations. With the historical urban flows and the flows of the super node, the model selects nodes with the similar spatial-temporal patterns directly.

In the imputation process, SATCNet captures the spatial and temporal correlations separately and STMCNet attains the complex spatial-temporal patterns. Next, we employ a linear combination function as follows to integrate them:

$$X_{c,t} = W_1 X_{a,t} + W_2 X_{m,t}, \quad (19)$$

where W_1, W_2 are learnable weight parameters. Finally, we aim to minimize the following objective function:

$$\mathcal{L}(X_{c,t}, Y_{c,t}, M_t) = \sum_{1}^{bz} \sum_{n=1}^{N} \|(X_{c,t}(n) - Y_{c,t}(n)) \odot M_t(n)\|_F^2 \quad (20)$$

where bz denotes batch size, N means sensor nodes and $Y_{c,t}$ is the ground truth of target urban flows. The pseudo-code is presented in Algorithm 1.

Algorithm 1 Pseudo-code of AST-CMCN

```
Initialization: SN_t, H_0, W, n ← 0, Message ← 0
Draw k_D samples from dataset {X_{u,t}(n), M(n), Y_{c,t}(n)}_{n=1}^{k_D}
while training loss has not converged do
  for n = 1, ..., k_D do
    for t = t_0, ..., t_k do
      r_t ← σ(Gate([X_{u,t}||H_{t-1}], A))
      u_t ← σ(Gate([X_{u,t}||H_{t-1}], A))
      c_t ← tanh(Gate([X_{u,t}||r_t ⊙ H_{t-1}], A))
      H_t ← u_t ⊙ H_{t-1} + (1 - u_t) ⊙ c_t
    end for
    X_{a,t} ← H_t
    for k = t_0, ..., t - 1 do
      Message ← Message + W_k · ODTrans(X_{u,k}, X_{u,t})
    end for
    Message ← Message + W_{k+1} · ODTrans(SN_t, X_{u,t})
    X_{m,t} ← Message
    X_{c,t} ← W_1 X_{a,t} + W_2 X_{m,t}
  end for
  Update D using adaptive moment estimation (Adam)
  ∇_D − Σ_{n=1}^{k_D} L(X_{c,t}(n), Y_{c,t}(n), M_t(n))
end while
```

5 EXPERIMENT

5.1 Experiment Setting

5.1.1 Datasets. We use two datasets for evaluation: Nanjingyby and PEMS08. As the raw datasets are complete, we artificially and randomly remove some values to mimic the missing values, including the random missing and the block missing. For random missing, we set urban flow values as 0 for the randomly selected regions or time slots. For block missing, we set the urban flow values of all regions as 0 in the randomly selected time slots. The details of the datasets are introduced as follows.

- **Nanjingyby** This dataset contains over 5 million crowd trip records in Nanjing Garden Expo Park from April 20 to June 30 in 2021. Each crowd trip includes id, regionid, time, latitude and longitude. We process the data with a time slot of 5 minutes. Then we partition the entire data into non-overlapping training, validation and testing sets by a ratio of 7 : 2 : 1.
- **PEMS08**[1] This dataset contains the traffic data in San Bernardino from July 1 to August 31 in 2016, with 170 detectors on 8 roads with a time slot of 5 minutes. This dataset contains three features: flow, occupy and speed. In this paper, we only employ the flow feature. We use the first 42 days data for training, the next 12 days data for validation, and the remaining data for testing.

5.1.2 Evaluation Metrics. We use two metrics Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) defined as follows to evaluate the model performance.

$$MAE = \frac{1}{T} \sum_{t=0}^{T} \|X_{c,t} - Y_{c,t}\| \quad (21)$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=0}^{T} (\|X_{c,t} - Y_{c,t}\|_F^2)} \quad (22)$$

where $X_{c,t}$ is the imputed value at time t and $Y_{c,t}$ is the corresponding ground truth.

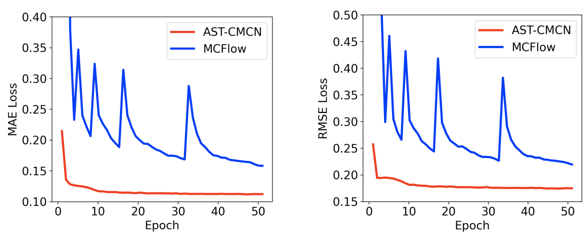
5.1.3 Baselines. We compare AST-CMCN with five baseline methods including statistics-based imputation methods, generative-based imputation methods and recent state-of-the-art attention-based imputation models. The details of the methods are introduced as follows.

- **MICE [6]** MICE is a statistic-based imputation method. It imputes the missing data through a series of iterative prediction models. In each iteration, other complete data in the dataset are used to estimate each incomplete variable in the dataset until the model converges.
- **GAIN [9]** GAIN is a generative-based imputation model. It utilizes the generative model GAN [7] to impute the missing data. In GAIN, the generator aims to accurately impute the

[1] https://github.com/wanhuaiyu/ASTGCN

Table 1: MAE and RMSE comparison among different methods. 25% **and** 40% **denote the missing ratio of urban flow data.**

model	Nanjingyby				PEMS08			
	25%		40%		25%		40%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MICE	0.2388	0.3429	0.2797	0.6046	0.3980	0.4435	0.4630	0.5185
GAIN	0.2153	0.2691	0.2554	0.3617	0.2487	0.3492	0.4201	0.4734
MCflow	0.1663	0.2408	0.2062	0.2779	0.1048	0.1423	0.1110	0.1531
MVTS	0.1572	0.2379	0.1991	0.2984	0.0986	0.1351	0.1012	0.1422
SAITS	0.1379	0.1976	0.1829	0.2773	0.0782	0.1007	0.0924	0.1255
AST-CMCN	**0.1171**	**0.1756**	**0.1344**	**0.2405**	**0.0541**	**0.0751**	**0.0674**	**0.0929**
Improvement	17.76%	12.53%	36.09%	15.30%	44.55%	34.09%	37.09%	35.09%



(a) MAE Loss curves. (b) RMSE Loss curves.

Figure 6: Loss curves of AST-CMCN and MCFlow on the Nanjingyby dataset.

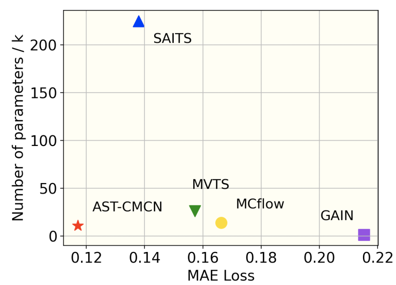


Figure 7: Parameter sizes and MAE losses of various methods.

missing data, and the discriminator is designed to distinguish the imputed data from the observed data.

- **MCflow [11]** MCflow is also a deep generative-based framework for imputation that leverages normalizing flow generative models and Monte-Carlo sampling. It addresses the causality dilemma that arises when training models with incomplete data by introducing an iterative learning strategy.
- **MVTS [13]** MVTS is a transformer-based framework for unsupervised representation learning of multivariate time series. Its pre-trained model can be used for downstream tasks such as missing value imputation.
- **SAITS [14]** SAITS is a method based on the self-attention mechanism for missing value imputation in multivariate time series. It learns missing values from a weighted combination of two DMSA blocks which explicitly capture both the temporal dependencies and feature correlations between time steps.

To study whether all the components of our model are useful, we also compare the full version AST-CMCN with its variants as follows.

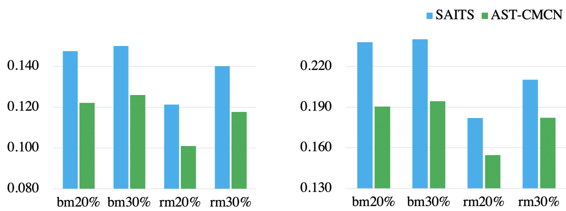
- **AST-CMCN-S** We replace Gate module in SATCNet with a non-linear function to examine whether spatial associations among sensor nodes can improve the model performance.
- **AST-CMCN-T** We remove the stacked GRUAtt modules from SATCNet and only employ one Gate module to study whether the temporal correlations are helpful for the urban flows imputation.
- **AST-CMCN-A** We remove the SATCNet module from AST-CMCN to test whether the separate spatial and temporal correlations are helpful for the data completion.
- **AST-CMCN-M** We train AST-CMCN without STMCNet module to examine the importance of the complex spatial-temporal correlations.
- **AST-CMCN-SN** We remove the super node from the network to test whether the new spatial-temporal patterns can be captured and are useful for this task.

5.1.4 Implementation Details. We implement our model as well as baselines with Pytorch framework on NVIDIA Tesla M40 GPU. We leverage Adam algorithm for gradient descent to perform training process with batch size $bz = 32$ and learning rate $lr = 1e - 4$. The model parameters are set as follows. The input urban flow data $\mathcal{X}_u$ is $6 \times N \times D$ for all datasets, where 6 is the previous time slot length, N is equals to 13 for Nanjingyby dataset and to 170 for PEMS08 dataset, D is the dimension of urban flow data. Note that when preprocessing the incomplete urban flow data, we generate the block missing and the random missing with the same ratio. For example, when the total missing ratio is 40%, block missing accounts for 20% and random missing accounts for 20%. And we normalize data the with Min-Max Normalization.

5.2 Comparison Result

To more extensively evaluate our model, we set the ratio of missing data as 25% and 40%, respectively, and make comparison with baselines in both cases on the two datasets.

Performance comparison. The performance result of different methods is shown in Table 1. The best results are highlighted with bold font, and the best results achieved by baselines are underlined. To more clearly show how much improvement our model



(a) MAE loss (b) RMSE loss

Figure 8: Imputation result on two data missing scenarios of SAITS and AST-CMCN models on Nanjingyby dataset. "rm20%" means 20% random missing and "bm20%" means 20% block missing.

achieves, we also present the improvement of AST-CMCN comparing to the best results achieved by baselines. It shows that our AST-CMCN model achieves the best result in all the cases. On Nanjingyby dataset, AST-CMCN performs best among all the methods and the improvement is significant. For example, when the missing ratio is 25%, MAE and RMSE of AST-CMCN are 0.1171 and 0.1756 respectively, improving by 17.76% and 12.53% comparing with the best results achieved by SAITS. MICE gives a bad result as it imputes incomplete urban flows by other available urban flow data, which ignores both spatial and temporal correlations. GAIN achieves the worst performance among all the deep learning methods. It is not surprising because it only employs deep generative model and does not design any components to consider spatial associations and temporal dependencies. One can see that MCflow improves performance comparing to MICE and GAIN model, as it uses Monte-Carlo sampling to capture inner distribution of urban flow data. However, it gives a bad performance because it still ignores the spatial-temporal correlations. MVTS and SAITS based on attention mechanism perform better than other baselines but still are inferior to AST-CMCN model. This is because the two methods are both designed for time series data, which cannot effectively capture the spatial associations. Similar to the results on the Nanjingyby dataset, AST-CMCN performs best among all the methods on PEMS08 dataset.

Model efficiency analysis Next, we further discuss the efficiency of our AST-CMCN model. We plot the MAE and RMSE loss curves of AST-CMCN and MCFlow model in Figure 6(a) and (b) over the Nanjingyby dataset to compare the computational efficiency of the models. The result shows that AST-CMCN converges faster and more stable than generative-based model MCFlow with less number of training epochs, which confirms the efficiency of our generative-free model AST-CMCN. To examine the superiority of AST-CMCN, we further compare all the models in terms of their parameter size and the corresponding MAE loss on the Nanjingyby dataset under 25% data missing ratio. The result is shown in Figure 7. One can see that AST-CMCN achieves the lowest MAE 0.1171 compared with other models, and at the same time its parameter size is small (only slightly larger than GAIN). It means that AST-CMCN achieves better result with a simpler model (less number of model parameters). GAIN has the least number of parameters, but its performance is not promising as the MAE is high.

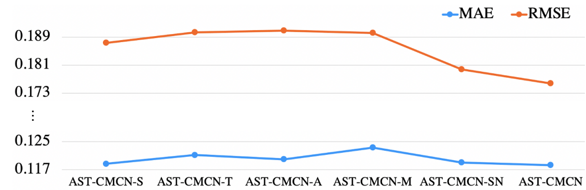
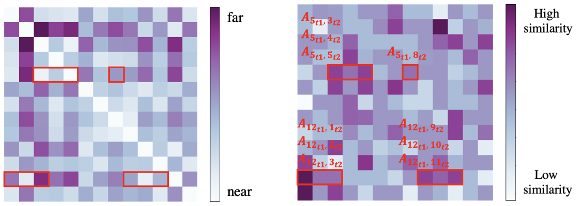


Figure 9: The comparison between AST-CMCN model and its variants.

Model effectiveness analysis on two data missing scenarios. To evaluate the effectiveness of AST-CMCN model on the two data missing scenarios, we examine the performance of AST-CMCN model on the two cases separately. We randomly remove some region values to generate the block missing and the random missing, separately. Specifically, we set block missing ratio as 20% and 30% and random missing as 20% and 30%. We compare AST-CMCN with the best baseline SAITS. The result is shown in Figure 8. One can see that AST-CMCN performs better than SAITS in all the cases, which can confirm the effectiveness of AST-CMCN on both block missing and random missing scenarios. It is not surprising that the MAE and RMSE loss of AST-CMCN model on 20%, 30% random missing perform better than on 20%, 30% block missing. It implies that blocking missing scenario is harder to impute than random missing scenario, because less information is available when all the data are missing in a time interval in block missing.

5.3 Ablation Study

We next conduct ablation study to examine whether all the designed components in AST-CMCN are all beneficial to the studied problem. We compare AST-CMCN with its variants including AST-CMCN-S, AST-CMCN-T, AST-CMCN-A, AST-CMCN-M and AST-CMCN-SN on the Nanjingyby dataset under 20% missing data ratio. The result is shown in Figure 9. One can see that AST-CMCN outperforms the five variants. It indicates that the Gate module, the GRUAtt module, SATCNet, STMCNet and the super node are all useful, and dropping any one of them will hurt the performance. One can also observe that STMCNet contributes most to the model performance improvement. Dropping STMCNet (AST-CMCN-M) will decrease the performance by up to 4.23%, 8.19% in terms of MAE loss and RMSE loss, respectively. It verifies STMCNet is especially important to the urban flows imputation. The imputation performance also drops significantly when the Gate module is removed (AST-CMCN-S), which demonstrates the importance of spatial correlations. Removing GREAtt module (AST-CMCN-T) will hurt the model performance. One can see in Figure 9 that the AST-CMCN-T performs worse than the AST-CMCN in both MAE loss and RMSE loss on Nanjingyby dataset, which proves the necessity of temporal relationship of urban flows. MAE and RMSE of AST-CMCN-A are both much larger than AST-CMCN, which means that the designed spatial and temporal completion network improves the task. Dropping the super node (AST-CMCN-SN) will hurt the model performance, since the MAE and RMSE loss of AST-CMCN-SN are larger than AST-CMCN. It demonstrates that the super node can capture the unseen spatial-temporal patterns outside the historical urban flows. AST-CMCN combines all these components together



(a) The adjacency matrix of nodes.

(b) The spatial-temporal correlations among regions at 2021/6/28 13:55 and 2021/6/28 14:20.

Figure 10: Visualization of the spatial-temporal correlations captured from STMCNet in Nanjing Garden Expo Park.

and achieves the best performance. Thus one can conclude that the well-designed components in AST-CMCN are all useful for the urban flow data imputation problem.

5.4 Visualization and Case study

In STMCNet, the Message Passing module can help capture the complex spatial-temporal relationships between sensor nodes in the network. To concretely show the ability of STMCNet, we visualize the complex spatial-temporal patterns captured by STMCNet in Figure 10. Figure 10(a) shows the static adjacency matrix of 13 sensor nodes based on geography in Nanjing Garden Expo Park, which can present the distance (near or far) between arbitrary two regions. Then, we choose the auxiliary urban flows at the time 2021/6/28 13:55 (t_1) and target urban flows at the time 2021/6/28 14:20 (t_2), and then visualize the correlations between them. In Figure 10(b), it is the correlations matrix between 13 nodes at two timestamps. For example, node 5 at t_1 has high similarity with nodes 3, 4, 5, 8 at t_2. According to the adjacency matrix of these nodes in Figure 10(a), one can see that these pairs of regions are geographically close. Thus, it is reasonable that they have the similar spatial-temporal patterns.

To further intuitively show the model performance, we visualize the imputation results with AST-CMCN model and the ground truth. A case study is visualized in Figure 11, showing the urban flow imputation in Nanjing Garden Expo Park. Figure 11(a) shows the imputation of the block missing flows at 2021/6/26 09:25. From the left figure of Figure 11(a), one can see that urban flows of all regions are unavailable (value is 0) because of communication errors. Then, after data imputation with AST-CMCN, the urban flows are completed as shown in the right figure in Figure 11(a). It shows that most of the imputed regions are rather accurate. For example, the impute value of Nantong garden is 10 and the ground truth is 11, the inferred flow of Zhenjiang garden is 8 and the ground truth is 10. One can see that there are inaccurate values, such as the urban flow data in Nanjing garden and Lianyungang garden. However, the imputation results are fairly accurate with small errors. Figure 11(b) represents the random missing imputation. One can see that urban flows in Wuxi garden and Suzhou garden are missing randomly. By SATCNet and STMCNet, we complete these two regions with relatively accurate values as shown in the right figure of Figure 11(b). This case study further verifies that our end-to-end AST-CMCN model can give the accurate complete urban flows for both random missing and block missing scenarios.

(a) Imputation the block missing flows at 2021/6/26 09:25.

(b) Imputation the random missing flows at 2021/6/26 09:40.

Figure 11: Visualization of the AST-CMCN urban flows imputation in Nanjing Garden Expo Park. “flows” denotes urban flow values, and “gt” denotes ground truth. White blocks denote missing urban flows and green ones are available value.

6 CONCLUSION

In this paper, we proposed a generative-free Attention-based Spatial-Temporal Combine and Mix Completion Network model to complete the missing urban flow data. AST-CMCN addressed three challenges, which were significant to the urban flow imputation process, including the effective generative-free model, the two types of missing data problems and the complex associations between spatial and temporal properties. In the future, it would be interesting to investigate whether we can design a better deep learning architecture to further improve the performance of the urban flow imputation.

7 ACKNOWLEDGEMENT

This research was funded by the National Science Foundation of China (No. 62172443, 62172034), the Science and Technology Major Project of Changsha (No. kh2202004), Guangdong Basic and Applied Basic Research Foundation (No.2021A1515012239), Hunan Provincial Natural Science Foundation of China (No. 2022JJ30053), the Beijing Natural Science Foundation (No. 4212021), CAAI-Huawei MindSpore Open Fund and the High Performance Computing Center of Central South University.

REFERENCES

[1] Kehua Su, Jie Li, and Hongbo Fu. Smart city and the applications. In *In the Proceedings of the 2011 international conference on electronics, communications and control (ICECC)*, pages 1028–1031, 2011.

[2] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):1–55, 2014.

[3] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. St-mvl: filling missing values in geo-sensory time series data. In *In the Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[4] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[5] Andrzej Cichocki and Anh-Huy Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 92(3):708–721, 2009.

[6] Ian R White, Patrick Royston, and Angela M Wood. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4): 377–399, 2011.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *In the Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2014.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *In the Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[9] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *In the Proceedings of the International Conference on Machine Learning (ICML)*, pages 5689–5698, 2018.

[10] Mengshi Qi, Jie Qin, Yu Wu, and Yi Yang. Imitative non-autoregressive modeling for trajectory forecasting and imputation. In *In the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12736–12745, 2020.

[11] Trevor W Richardson, Wencheng Wu, Lei Lin, Beilei Xu, and Edgar A Bernal. Mcflow: Monte carlo flow models for data imputation. In *In the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14205–14214, 2020.

[12] Weibin Zhang, Pulin Zhang, Yinghao Yu, Xiying Li, Salvatore Antonio Biancardo, and Junyi Zhang. Missing data repairs for traffic flow with self-attention generative adversarial imputation net. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[13] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *In the Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (SIGKDD)*, pages 2114–2124, 2021.

[14] Wenjie Du, David Côté, and Yan Liu. Saits: Self-attention-based imputation for time series. *arXiv preprint arXiv:2202.08516*, 2022.

[15] Senzhang Wang, Jiannong Cao, and Philip Yu. Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[16] Senzhang Wang, Jiannong Cao, Hao Chen, Hao Peng, and Zhiqiu Huang. Seqstgan: Seq2seq generative adversarial nets for multi-step urban crowd flow prediction. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 6(4):1–24, 2020.

[17] Peng Xie, Tianrui Li, Jia Liu, Shengdong Du, Xin Yang, and Junbo Zhang. Urban flow prediction from spatiotemporal data using machine learning: A survey. *Information Fusion*, 59:1–12, 2020.

[18] Sangsoo Lee and Daniel B Fambro. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record*, 1678(1):179–188, 1999.

[19] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *In the Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *In the Proceedings of the Thirty-first AAAI conference on Artificial Intelligence (AAAI)*, 2017.

[22] Zheyi Pan, Wentao Zhang, Yuxuan Liang, Weinan Zhang, Yong Yu, Junbo Zhang, and Yu Zheng. Spatio-temporal meta learning for urban traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[23] Senzhang Wang, Hao Miao, Hao Chen, and Zhiqiu Huang. Multi-task adversarial spatial-temporal networks for crowd flow prediction. In *In the Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1555–1564, 2020.

[24] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. Multi-stage attention spatial-temporal graph networks for traffic prediction. *Neurocomputing*, 428:42–53, 2021.

[25] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. *In the Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[26] Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *In the Proceedings of the International Conference on Machine Learning (ICML)*, pages 4413–4423, 2019.

[27] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *In the Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[28] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *In the Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1724–1734, 2014.

[29] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *In the Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.