

# Irregular Traffic Time Series Forecasting Based on Asynchronous Spatio-Temporal Graph Convolutional Networks

Anonymous Author(s)

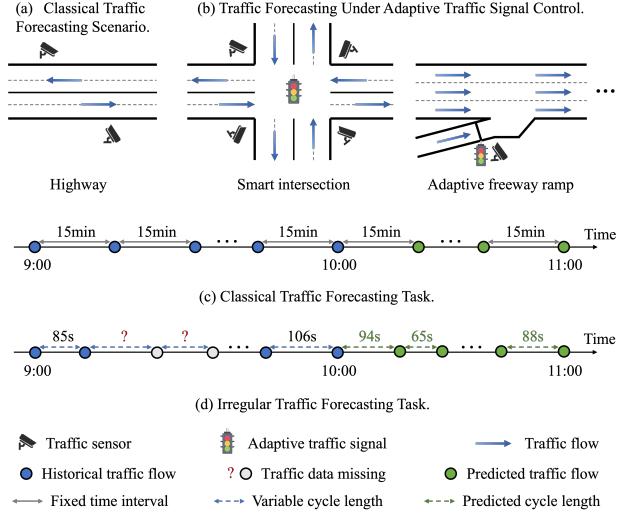
## ABSTRACT

Accurate traffic forecasting is crucial for the development of Intelligent Transportation Systems (ITS), playing a pivotal role in modern urban traffic management. Traditional forecasting methods, however, struggle with the irregular traffic time series resulting from adaptive traffic signal controls, presenting challenges in asynchronous spatial dependency, irregular temporal dependency, and predicting variable-length sequences. To this end, we propose an Asynchronous Spatio-tEmporal graph convolutional nETwoRk (ASEER) tailored for irregular traffic time series forecasting. Specifically, we first propose an Asynchronous Graph Diffusion Network to capture the spatial dependency between asynchronously measured traffic states regulated by adaptive traffic signals. After that, to capture the temporal dependency within irregular traffic state sequences, a personalized time encoding is devised to embed the continuous time signals. Then, we propose a Transformable Time-aware Convolution Network, which adapts meta-filters for time-aware convolution on the irregular sequences. Additionally, a Semi-Autoregressive Prediction Network, comprising a state evolution unit and a semi-autoregressive predictor, is designed to predict variable-length traffic sequences effectively and efficiently. Extensive experiments on two constructed real-world datasets demonstrate the effectiveness of ASEER compared with eleven competitive baseline approaches in six metrics. Our implementation is provided at <https://anonymous.4open.science/r/itfcode-5C4C/>.

## 1 INTRODUCTION

Recent years have witnessed significant advancements in traffic forecasting, which plays a pivotal role in underpinning Intelligent Transportation Systems (ITS) [19], facilitating emergency response and management [28], and is integral to the development of autonomous driving [17]. In particular, timely and accurate traffic forecasting is of great importance to help the Intelligent Traffic Signal Control Systems (ITSCS) to anticipate future traffic state variations, thereby providing crucial insights to support the systematic analysis, informed decisions, and optimal control optimization of ITSCS to enhance the overall transportation system efficiency [38].

In practice, the traffic dynamics of the road network is jointly decided by the vehicles on the road and the intervention of traffic signals, *e.g.*, intersection traffic lights, ramp metering lights, and lane allocation signals [5, 37, 39]. On the one hand, the traffic signal adaptively adjusts its control cycles in response to real-time traffic flow variations [39]. On the other hand, traffic flows are dynamically regulated by these adaptive signal control strategies with varying cycle lengths. As a result, the urban traffic states, entangling both length-varying traffic signal cycles and the corresponding traffic flows, exhibit significant irregularity and render more complex traffic dynamics, as depicted in Figure 1.



**Figure 1: The distinction between classical traffic forecasting on the highway and irregular traffic forecasting under adaptive traffic signal control. The forecasting task aims to predict future traffic variations (10:00-11:00) based on past observations (9:00-10:00).**

However, existing studies [15, 21, 42, 45] on urban traffic forecasting primarily focus on capturing spatiotemporal dependencies among geo-distributed time series with evenly spaced temporal variables, largely overlooking the irregularity of traffic time series induced by the interplay between traffic dynamics and intelligent traffic control policies. These approaches model spatiotemporal dependencies and forecast traffic variations within fixed time intervals, which are misaligned with the traffic control cycles, leading to substantial fluctuations and inaccuracies in understanding traffic dynamics and forecasting outcomes. In this study, we investigate the irregular traffic time series forecasting (*a.k.a.* irregular traffic forecasting) task, aiming to predict a variable-length sequence of traffic states, encompassing traffic signal cycle lengths and the corresponding traffic flows, in the future time window based on incomplete historical traffic state sequences characterized by variable time intervals.

It is a non-trivial task due to the following three major challenges: (1) *Asynchrony in spatial dependency modeling*. Traffic time series has obvious spatial dependency due to the traffic state's diffusion nature on road network [21]. However, due to the time-misaligned traffic signal cycles (diverged cycle beginning time and length) between sensors, their traffic state measurements under the adaptive control policy would be observed asynchronously. Such asynchrony hinders correlating and integrating these sensors' traffic states synchronously favored by classical traffic forecasting methods [21, 45],

presenting a substantial challenge to model their spatial dependency. (2) *Irregularity in temporal dependency modeling*. The future traffic states are correlated with their historical values. Unlike previous traffic forecasting research [21, 45] that deals with regular traffic time series, we need to handle irregular traffic state sequences characterized by variable time intervals between successive measurements. These irregularities stem from fluctuating cycle lengths and data missing in sensor measurements, challenging classical traffic forecasting methods to capture the underlying temporal dynamics and dependencies precisely. (3) *Variable-length sequence to be predicted*. Our goal is to predict the complete traffic state sequences in a future time window (e.g., the next hour). However, due to signal cycle lengths varying across different sensors and times, the lengths of sequences to be predicted also vary. While an autoregressive prediction model may seem plausible for predicting sequences of variable lengths, significant error accumulation and poor prediction efficiency issues are pronounced with longer sequences [25, 26]. This presents a significant challenge in effectively and efficiently forecasting variable-length traffic state sequences.

To tackle the above challenges, we present an Asynchronous Spatio-tEmporal graph convolutional nEtwoRk (ASEER) for irregular traffic forecasting. Specifically, building on a traffic diffusion graph where traffic sensors are nodes and edges are defined by geographical proximity and road network reachability, we propose an Asynchronous Graph Diffusion Network to model the spatial dependencies among nodes with time-misaligned traffic state measurements. It allows each node to asynchronously diffuse its traffic measurements to neighbors and store received traffic information with a message buffer. The stored messages are then integrated through an asynchronous graph convolution for the spatial node representation. To capture temporal dependencies in irregular sequences, a learnable personalized time encoding is first devised to embed the continuous times of traffic measurements. Then, we propose a Transformable Time-aware Convolution Network that learns meta-filters to derive time-aware convolution filters with transformable filter sizes, which are applied for efficient temporal convolution on the irregular sequences. Lastly, we design a Semi-Autoregressive Prediction Network to iteratively predict variable-length traffic state sequences effectively and efficiently. It incorporates a state evolution unit to evolve traffic's hidden state with elapsed time and a semi-autoregressive predictor to predict a sequence of consecutive traffic states at each prediction step.

Our major contributions can be summarized as follows: (1) We investigate a novel irregular traffic forecasting problem, which imposes three critical new challenges for traffic forecasting from spatial, temporal, and predicted sequence length perspectives. (2) We propose an Asynchronous Graph Diffusion Network to model spatial dependency among asynchronous traffic time series data. (3) We propose a Transformable Time-aware Convolution Network with personalized time encoding to efficiently capture temporal dependency within irregular traffic time series. (4) We design a Semi-Autoregressive Prediction Network to empower effective and efficient prediction for variable-length traffic time series. (5) We meticulously collect and develop two novel real-world datasets of irregular traffic time series from two leading pilot cities for ITSCS in China, and establish a systematic evaluation scheme comprising six metrics, setting a new benchmark in the field. Extensive experiments

demonstrate the superiority of ASEER compared with eleven competitive baseline approaches. Both source code and datasets will be opened to inspire future studies and foster significant progress in related areas.

## 2 PRELIMINARIES

Consider a set of  $N$  traffic sensors, denoted as  $\mathbf{V} = \{v^1, v^2, \dots, v^N\}$ , positioned on lanes governed by adaptive traffic signals, e.g., lanes connecting to smart intersections. Each sensor gathers real-time traffic data specific to a lane.

**Definition 1: Traffic State Measurement.** The  $n$ -th chronological traffic state measurement of a sensor  $v^i$  is defined as  $x_n^i = \langle p_n^i, f_n^i \rangle$ , where  $p_n^i$  denotes the traffic signal cycle length under the adaptive control, and  $f_n^i$  is the traffic flow during this signal cycle. We further define  $b_n^i$  and  $t_n^i$  as the beginning and end timestamps (in second) of this signal cycle, and we have  $t_n^i = b_n^i + p_n^i - 1$ .

As traffic signal cycles occur consecutively in real-world scenarios, we have  $b_{n+1}^i = t_n^i + 1$  in case of no missing traffic states between  $x_n^i$  and  $x_{n+1}^i$ . Due to the unpredictable systematic failures of sensors, there could be multiple traffic states missing between two successive observed measurements.

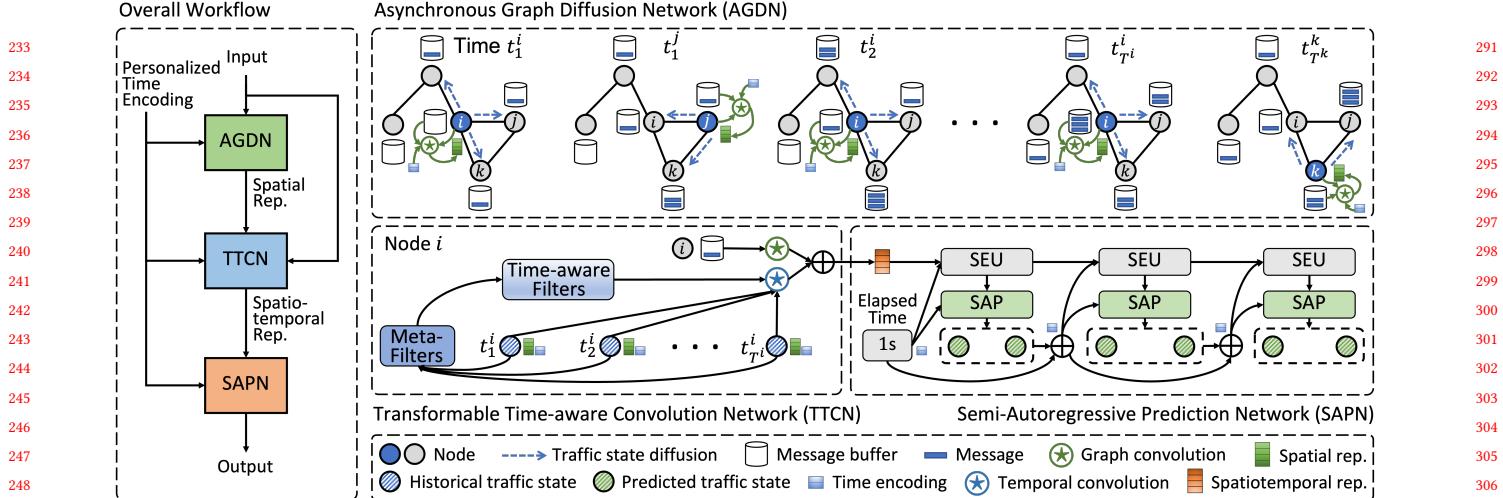
**Problem: Irregular Traffic Forecasting.** Given a historical time window  $\mathcal{T}$ , e.g., one hour, before current timestamp  $t$ , and a set of historical traffic state measurements  $\mathbf{X}_{[t-\mathcal{T}+1:t]} = \{[x_n^i]_{n=1}^{T^i}\}_{i=1}^N$  of all sensors  $\mathbf{V}$  obtained during  $\mathcal{T}$ , where  $T^i$  is the number of observed historical measurements of  $v^i$ , our problem is to predict the complete traffic states  $\mathbf{Y}_{[t+1:t+\tau]} = \{[x_n^i]_{n=T^i+1}^{T^i+L^i}\}_{i=1}^N$  for all sensors in the next  $\tau$  time window, e.g., the next hour, formalized as:

$$\mathcal{F}(\mathbf{X}_{[t-\mathcal{T}+1:t]}) \longrightarrow \mathbf{Y}_{[t+1:t+\tau]}, \quad (1)$$

where  $L^i$  is the number of ground truth traffic states of  $v^i$  during the predicted time window,  $\mathcal{F}(\cdot)$  represents the forecasting model we aim to learn. Note that we use the subscript  $[a:b]$  to indicate a time window spanning from timestamp  $a$  to  $b$ .

## 3 METHODOLOGY

**Framework Overview.** Figure 2 shows the framework overview of ASEER, which consists of three major components. Specifically, Asynchronous Graph Diffusion Network (AGDN) models asynchronous spatial dependency based on a traffic diffusion graph, where nodes are represented by sensors and edges are constructed based on sensors' geographical proximity and road network reachability. When a node has a traffic state measurement, AGDN asynchronously diffuses the node's traffic measurement to its neighbors, which receive and store the diffused traffic state into their message buffers. Next, the node performs an asynchronous graph convolution to obtain spatial representation through attentively integrating the stored traffic messages, and then the buffer will be cleared. After that, a Transformable Time-aware Convolution Network (TTCN) is adopted to model the temporal dependency within irregular traffic state sequences. TTCN learns meta-filters to derive time-aware convolution filters with transformable filter sizes based on spatial representations obtained from AGDN and traffic measurements along with personalized time encoding. Then the derived time-aware convolution filters are applied for efficient temporal convolution on irregular traffic state sequences to acquire



**Figure 2:** The framework overview of ASER, which consists of three major components: Asynchronous Graph Diffusion Network (AGDN), Transformable Time-aware Convolution Network (TTNCN), and Semi-Autoregressive Prediction Network (SAPN). The traffic states are first inputted to AGDN to obtain spatial representations, which are incorporated by TTNCN to acquire the spatiotemporal representations. After that, SAPN predicts the variable-length traffic state sequence based on the spatiotemporal representations. Throughout the entire process, personalized time encoding is used to embed continuous time.

the spatiotemporal representation for each node. Finally, a Semi-Autoregressive Prediction Network (SAPN) is devised to iteratively predict variable-length traffic state sequences. In each prediction step, a State Evolution Unit (SEU), whose hidden state is initialized by spatiotemporal representations, is first introduced to evolve each node's future traffic hidden state with the elapsed time, then a Semi-Autoregressive Predictor (SAP) is adopted to predict a sequence of consecutive traffic states based on both evolutionary, initial traffic hidden states, and predicted elapsed time.

### 3.1 Asynchronous Spatial Dependency Modeling

Previous traffic forecasting studies model spatial dependency by introducing graph neural networks to synchronously diffuse and aggregate the time-aligned traffic state measurements between different sensor nodes [21, 45]. However, in our problem, the observed traffic state measurements of different sensors cannot be aligned due to the distinct timestamps of their traffic signal cycles and the data missing issue, which causes severe asynchrony in spatial dependency modeling.

To this end, by linking sensors via a traffic diffusion graph, we propose an Asynchronous Graph Diffusion Network (AGDN), as illustrated in Figure 2, to model the asynchronous spatial dependency between the time-misaligned traffic measurements. The key idea of AGDN is that each node asynchronously diffuses its traffic state measurement to the adjacent nodes once it's observed. Then, the adjacent nodes receive and store traffic state from others into a message buffer of itself. Next, each node will integrate its stored traffic messages via an asynchronous graph convolution operation. We detail it below.

**Diffusion Graph Construction.** To model spatial dependency between traffic sensors, we construct a traffic diffusion graph  $\mathcal{G} = (\mathcal{V}, \mathcal{X}_{\mathcal{V}}, \mathcal{E}, \mathcal{X}_{\mathcal{E}})$ , where the graph nodes  $\mathcal{V} = \mathbf{V}$  represents a set of sensors,  $\mathcal{X}_{\mathcal{V}} = \mathbf{X}_{[t-\mathcal{T}+1:t]}$  denotes features of nodes  $\mathcal{V}$ ,  $\mathcal{E}$  are a set

of edges indicating proximity between nodes, and  $\mathcal{X}_{\mathcal{E}}$  are features in edges  $\mathcal{E}$ . Specifically, we define proximity  $e^{ij} \in \mathcal{E}$  between  $v^i$  and  $v^j$  as:

$$e^{ij} = \begin{cases} 1, & \text{if } \text{dist}(v^i, v^j) < \epsilon, i \neq j \\ 0, & \text{if otherwise} \end{cases}, \quad (2)$$

where  $\text{dist}(v^i, v^j)$  denotes geographical distance between node  $v^i$  and  $v^j$ ,  $\epsilon$  is a threshold, and there is no self-loop for each node. We also define some edge features  $x_e^{ij} \in \mathcal{X}_{\mathcal{E}}$  between nodes  $v^i$  and  $v^j$ , including geographical distance and the direct reachability in the lane-level road network. Note that it's not limited to geographical proximity and reachability, other graph construction approaches can also be embraced.

**Asynchronous Diffusion and Storage.** Assume a traffic state measurement  $x_{n^-}^j$  of node  $v^j$  is observed at timestamp  $t_{n^-}^j$ , then  $v^j$  will diffuse  $x_{n^-}^j$  as a traffic message to its adjacent nodes  $v^i \in \mathcal{N}_j$  in terms of edges  $\mathcal{E}$ , which can be more formally denoted as:

$$\text{AsynDiff}\left(v^j \xrightarrow{x_{n^-}^j} \{v^i : \forall v^i \in \mathcal{N}_j\}\right). \quad (3)$$

For each node  $v^i \in \mathcal{N}_j$ , it receives the traffic message  $x_{n^-}^j$  and then stores it into its message buffer  $\mathcal{B}^i$  for later use:

$$\text{Store}\left(x_{n^-}^j \longrightarrow \{\mathcal{B}^i : \forall v^i \in \mathcal{N}_j\}\right). \quad (4)$$

Since the timestamps of traffic state measurements are misaligned for different nodes, the traffic messages diffusion and storage processes perform in an asynchronous way.

**Asynchronous Graph Convolution.** An immediate problem is how to exploit traffic messages stored in the message buffer to enhance each node's spatial perception. We achieve this by enforcing each node  $v^i$  asynchronously integrates the traffic messages in its message buffer  $\mathcal{B}^i$  via an asynchronous graph convolution operation, which is performed once a traffic state measurement  $x_n^i$  is observed in node  $v^i$ .

Specifically, we first employ  $x_n^i$  to query the message buffer  $\mathcal{B}^i$  for the proximity weights computation with each traffic message

349  $x_{n^-}^j \in \mathcal{B}^i$  via the following attention operation:

$$350 \quad \alpha_{nn^-} = \frac{\exp(\beta_{nn^-})}{\sum_{x_{n'}^j \in \mathcal{B}^i} \exp(\beta_{nn'}^j)}, \quad (5)$$

$$353 \quad \beta_{nn^-} = \mathbf{v}^\top \tanh \left( \mathbf{W}_a [x_n^i \oplus x_{n^-}^j \oplus \phi^i(t_n^i - t_{n^-}^j) \oplus x_e^{ij}] \right),$$

354 where  $\oplus$  indicates concatenation operation,  $\mathbf{v}$  and  $\mathbf{W}_a$  are learnable parameters, and  $\phi^i(\cdot)$  is a learnable time encoding function to embed cycle-related patterns for each node that will be detailed in the next section.

355 Once the proximity weights are obtained, we asynchronously integrate node's stored traffic messages received from neighbors via 356 an attentive graph convolution to obtain the spatial representation:

$$362 \quad \tilde{h}_n^i = \text{MLP} \left( \sum_{x_{n^-}^j \in \mathcal{B}^i} \alpha_{nn^-} \cdot [x_{n^-}^j \oplus \phi^i(t_n^i - t_{n^-}^j) \oplus x_e^{ij}] \right), \quad (6)$$

363 where MLP represents a multi-layer perceptron. It's noteworthy 364 that after each asynchronous graph convolution operation on  $\mathcal{B}^i$ , 365 all the traffic messages in it will be cleared. It indicates that each 366 node only integrates adjacent traffic messages from its last traffic 367 measurement's timestamp to the current measurement's timestamp 368  $t_n^i$ , which guarantees each message is utilized exactly once to avoid 369 redundant information and computation.

370 Define  $x_{T^i}^i$  as the last observed traffic state measurement of 371 node  $v^i$  during historical time window  $\mathcal{T}$ . There could be some 372 messages received and stored in the message buffer  $\mathcal{B}^i$  after the 373 timestamp  $t_{T^i}^i$ . Thus, we perform a similar asynchronous graph 374 convolution operation for these remaining messages by adding 375 a virtual measurement  $\bar{x}_{T^i}^i$  at timestamp  $t_{T^i}^i$  without traffic state 376 values. The obtained spatial representation is denoted as  $\bar{h}_{T^i}^i$ .

### 3.2 Irregular Temporal Dependency Modeling

381 Convolutional Neural Network (CNN) [20] is widely applied to classical traffic forecasting tasks for its both efficiency and effectiveness 382 in temporal dependency modeling [10, 18, 42, 45]. However, applying CNN to our task faces two problems. First, CNN fails to directly process irregular traffic sequences with variable sequence lengths. 383 Second, CNN is incompetent to model temporal dependency in the sequence with varying time intervals, as its filter parameters are 384 fixed and cannot adaptively adjust according to the varying time 385 intervals between successive data points.

386 To tackle the above problems, we propose a Transformable Time- 387 aware Convolution Network (TTCN) that enables to model irregular 388 sequences with transformable time-aware convolution filters 389 which learns a meta-filter to derive the time-aware convolution 390 filter with transformable filter size so that enables to efficiently 391 model temporal dependency in the irregular traffic state sequence, 392 and further devise a personalized time encoding function to embed 393 the unique cycle-related patterns for each node. Specifically, given 394 the historical time window  $\mathcal{T}$  before  $t$ , for each node  $v^i$ , we first 395 concatenate the traffic state measurement  $x_n^i$  during  $\mathcal{T}$  with the 396 corresponding spatial representation  $\tilde{h}_n^i$  and the encoding of time 397 intervals  $\phi^i(t_{T^i}^i - t_n^i)$  from timestamp  $t_n^i$  to timestamp  $t_{T^i}^i$  of the 398 last traffic state measurement:

$$405 \quad z_n^i = [x_n^i \oplus \tilde{h}_n^i \oplus \phi^i(t_{T^i}^i - t_n^i)]. \quad (7)$$

407 **3.2.1 Personalized Time Encoding.** The desired time encoding 408 should not only indicate the absolute time interval but also imply 409 the unique cycle-related patterns of traffic dynamics in different 410 nodes. For example, a time interval may signify a distinct number 411 of traffic signal cycles for different sensors, which is important for 412 temporal dependency modeling, especially when the time interval 413 spans multiple missing traffic states. Inspired by the positional 414 encoding in Transformer [35], we introduce a personalized time 415 encoding by adopting a learnable trigonometric function to embed 416 the time interval  $\Delta t$  for each node:

$$417 \quad \phi_p^i(\Delta t)[s] = \begin{cases} \Delta t, & \text{if } s = 0 \\ \sin(\omega_k^i \Delta t), & \text{if } s = 2k + 1 \\ \cos(\omega_k^i \Delta t), & \text{if } s = 2k + 2 \end{cases}, \quad (8)$$

418 where the above equation denotes  $s$ -th element of the time encoding 419  $\phi_p^i(\Delta t) \in \mathbb{R}^{d_{\phi}+1}$ , and  $\omega_k^i$  are learnable parameters to indicate the 420 cyclical characteristics of this function. Each node has an individual 421 time encoding function with separate parameters so that can learn 422 its unique cycle-related patterns.

423 Due to the data missing problem, some nodes may have too 424 sparse measurement data to learn a satisfactory unique time encoding 425 function. Hence, we also jointly learn a generic time encoding 426  $\phi_g(\Delta t)$ , which has a similar function expression to Eq. (8) but is 427 shared by all nodes. Then, we introduce a learnable weight  $\lambda_i$  for 428 each node to adaptively integrate the above two time encoding:

$$429 \quad \phi^i(\Delta t) = (1 - \exp(-\lambda_i^2)) \cdot \phi_p^i(\Delta t) + \exp(-\lambda_i^2) \cdot \phi_g(\Delta t). \quad (9)$$

430  $\lambda_i$  is initialized to be close to zero so that the nodes with limited or 431 even no available data can weigh more on generic time encoding.

432 **3.2.2 Transformable Time-aware Convolution Network.** In 433 this section, we assume all the following operations are performed 434 on node  $v^i$ , thus we omit the superscript  $i$  to ease the presentation. 435 We first define  $\mathbf{z}_{[t-\mathcal{T}+1:t]} = \{z_1, \dots, z_T\}$  and  $T$  as the sequence 436 length. As illustrated in Figure 2, we then leverage a meta-filter to 437 derive the time-aware convolution filter with dynamic parameters 438 and transformable filter size  $T$  based on sequence inputs, formulated 439 as:

$$440 \quad \mathbf{f}_d = [\text{Norm}(\mathbf{F}_d(z_1)), \dots, \text{Norm}(\mathbf{F}_d(z_T))],$$

$$441 \quad \text{Norm}(\mathbf{F}_d(z_n)) = \frac{\exp(\mathbf{F}_d(z_n))}{\sum_{z_{n'} \in \mathbf{z}_{[t-\mathcal{T}+1:t]}} \exp(\mathbf{F}_d(z_{n'}))}, \quad (10)$$

442 where  $\mathbf{f}_d \in \mathbb{R}^{T \times D_{in}}$  is the derived filter for  $d$ -th feature map, and  $\mathbf{F}_d$  443 denotes the meta-filter that can be instantiated by learnable neural 444 networks. We normalize the derived filter parameters along the 445 temporal dimension to ensure consistent scaling of the convolution 446 results for variable-length sequences.

447 With  $D$  filters derived according to Eq. (10), we obtain the traffic 448 sequence representation  $h_T \in \mathbb{R}^D$  via the following temporal 449 convolution operation:

$$450 \quad h_T = [\mathbf{z}_{[t-\mathcal{T}+1:t]} \star \mathbf{f}_1, \dots, \mathbf{z}_{[t-\mathcal{T}+1:t]} \star \mathbf{f}_D],$$

$$451 \quad \mathbf{z}_{[t-\mathcal{T}+1:t]} \star \mathbf{f}_d = \sum_{n=1}^T \mathbf{f}_d[n]^\top \mathbf{z}_{[t-\mathcal{T}+1:t]}[n], \quad (11)$$

452 where  $\star$  denotes the convolution operation. Then we attain the 453 overall spatiotemporal representation for each node via the 454 representations integration:  $\mathbf{h}_T = h_T + \bar{h}_T$ .

Compared to CNN, TTCN has several advantages in modeling sequence's temporal dependency with irregular time intervals. First, the derived filter is transformable according to sequence length, which enables it to adaptively process variable-length sequences. Moreover, it can derive tailored parameterized filters for sequences with varying time intervals or other characteristics. It is worth noting that as the learnable parameters of meta-filter are independent of sequence length, TTCN is allowed to directly model the long-term temporal dependency through an arbitrarily large-size convolution filter without increasing any filter parameters.

### 3.3 Variable-Length Traffic Sequence Prediction

Our goal is to predict the complete traffic state sequences, including a sequence of traffic signal cycle lengths and the corresponding traffic flows, for all nodes in a future time window. However, the sequences to be predicted have variable lengths in terms of the differences in sensors, time windows, or prediction algorithms, and the sequence lengths cannot be known in advance. While an autoregressive prediction model that iteratively predicts the next step's value based on previously predicted values seems feasible for the variable-length sequence prediction, the prediction for long sequence can lead to severe error accumulation and poor prediction efficiency issues in this approach [25, 26].

To tackle the above problems, as displayed in Figure 2, we design a Semi-Autoregressive Prediction Network (SAPN) to iteratively predict sub-sequences until the complete sequence meets the requirements of the task, which not only enables variable-length sequence prediction in an efficient way but also mitigates the error accumulation issue for long sequence prediction. SAPN consists of a state evolution unit to evolve each node's traffic hidden state with the elapsed time and a semi-autoregressive predictor to predict a sequence of consecutive traffic states in each prediction step. Since prediction processes are the same for all nodes, we omit the superscript  $i$  to ease the presentation as well.

To be specific, we employ the spatiotemporal representation  $\mathbf{h}_T$  acquired from AGDN and TTCN as the initial traffic hidden state. In each prediction step, a semi-autoregressive predictor predicts a sequence of consecutive traffic states based on the evolutionary and initial traffic hidden states, as well as the predicted elapsed time encoding, formulated as:

$$[\hat{p}_n, \hat{u}_n]_{n=T+m\xi+1}^{T+(m+1)\xi} = \text{SAP}\left([\hat{\mathbf{h}}_{T+m\xi+1} \oplus \mathbf{h}_T \oplus \phi(\hat{\delta}_{T+m\xi+1})]\right), \quad (12)$$

where  $\xi$  is the prediction step size,  $m \geq 0$  denotes  $m$ -th prediction step,  $[\hat{p}_n, \hat{u}_n]_{n=T+m\xi+1}^{T+(m+1)\xi}$  respectively represent a sequence of consecutive cycle lengths and unit time (per second) traffic flows, and  $\hat{\delta}_{T+m\xi+1}$  indicates the elapsed time to the timestamp  $t_T$  of sequence's last observed measurement.  $\hat{\delta}_{T+m\xi+1}$  is initialized to 1 when  $m = 0$ , and iteratively updates based on the accumulation of predicted cycle lengths:

$$\hat{\delta}_{T+(m+1)\xi+1} = \hat{\delta}_{T+m\xi+1} + \sum_{k=1}^{\xi} \hat{p}_{T+m\xi+k}. \quad (13)$$

Since the underlying traffic state is actually dynamically evolving with passage of time, we introduce a state evolution unit to learn to evolve each node's traffic hidden state with the elapsed time:

$$\hat{\mathbf{h}}_{T+m\xi+1} = \text{SEU}\left(\hat{\mathbf{h}}_{T+(m-1)\xi+1}, \phi(\hat{\delta}_{T+(m-1)\xi+1})\right), \quad (14)$$

where  $\hat{\delta}_{T+(m-1)\xi+1} = 1$  and  $\hat{\mathbf{h}}_{T+(m-1)\xi+1} = \mathbf{h}_T$  if  $m = 0$ , otherwise  $\hat{\delta}_{T+(m-1)\xi+1} = \sum_{k=1}^{\xi} \hat{p}_{T+(m-1)\xi+k}$ , representing the elapsed time to last update of the traffic hidden state. Next, we can obtain the corresponding traffic flows of predicted traffic signal cycles by multiplying the predicted unit time traffic flows with cycle lengths:

$$[\hat{p}_n]_{n=T+m\xi+1}^{T+(m+1)\xi} = [\hat{u}_n]_{n=T+m\xi+1}^{T+(m+1)\xi} \odot [\hat{p}_n]_{n=T+m\xi+1}^{T+(m+1)\xi}, \quad (15)$$

where  $\odot$  denotes Hadamard product. By iteratively performing the above prediction step until the predicted sequence covers the required time window, we can derive the variable-length traffic state sequence we expect.

Compared to autoregressive models, our SAPN predicts a variable-length sequence with fewer prediction steps, which improves prediction efficiency and may reduce the risks of causing prediction error accumulation. It's not hard to see that both the autoregressive and non-autoregressive prediction models can be regarded as a special case of semi-autoregressive model when the prediction step size is set to one or the length of sequence. Thus, our SAPN can also be considered as incorporating both strengths of autoregressive and non-autoregressive prediction models to predict variable-length sequences. In the implementation, we instantiate SAP via MLP and SEU via Gated Recurrent Unit [7] as its mechanism aligns well with the recurrent update for traffic hidden state.

### 3.4 Model Training

Due to the data missing problem, we design three masked losses to train our model. The first loss is introduced to optimize the traffic signal cycle length forecasting via the masked Mean Absolute Error (MAE):

$$\mathcal{L}_p = \frac{1}{N \times L_1^i} \sum_{i=1}^N \sum_{l=1}^{L_1^i} \left| \hat{p}_{T^i+l}^i - p_{T^i+l}^i \right| \times \zeta_{T^i+l}^i, \quad (16)$$

where  $T^i$  is the length of the historical measurement sequence of node  $v^i$  and  $L_1^i$  is the length of the ground truth traffic state sequence of  $v^i$ .  $\zeta_{T^i+l}^i$  is a mask item, which equals zero if the ground truth value  $p_{T^i+l}^i$  is missing, otherwise it equals one, and  $L_1^i$  denotes the number of nonzero mask items for each node.

To further mitigate error accumulation in cycle length prediction, we additionally introduce a timing loss to improve the accuracy of predicted elapsed time accumulated by cycle lengths:

$$\mathcal{L}_\delta = \frac{1}{N \times L_1^i} \sum_{i=1}^N \sum_{l=1}^{L_1^i} \left| \hat{\delta}_{T^i+l}^i - \delta_{T^i+l}^i \right| \times \zeta_{T^i+l}^i. \quad (17)$$

Similarly, we introduce a masked MAE loss to optimize the corresponding traffic flow prediction for each traffic signal cycle:

$$\mathcal{L}_f = \frac{1}{N \times L_1^i} \sum_{i=1}^N \sum_{l=1}^{L_1^i} \left| \hat{u}_{T^i+l}^i \times p_{T^i+l}^i - f_{T^i+l}^i \right| \times \zeta_{T^i+l}^i. \quad (18)$$

Since traffic flow prediction is also based on cycle lengths, to avoid disturbance from the error of predicted cycle lengths, we use the ground truth cycle lengths to calculate the corresponding traffic flows in the training phase.

Consequently, ASEER aims to jointly minimize an overall objective that combines the above three masked losses:

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_\delta + \mathcal{L}_f. \quad (19)$$

**Table 1: Statistics of datasets.**

Description	ZHUZHOU	BAODING
# of measurements	19,824,504	13,093,975
# of sensors	620	264
Time range	2022/07/20-2022/10/02	2021/12/01-2022/02/25
Missing period ratio	44.2%	27.2%
Average / maximal ground truth sequence length to be predicted	57 / 213	64 / 155

## 4 EXPERIMENTS

### 4.1 Experimental Setup

**Datasets.** We conduct experiments on two real-world datasets, ZHUZHOU and BAODING, which represent two major pilot cities for ITSCS and autonomous driving in China. Both datasets consist of a set of entrance lanes connecting to smart intersections and traffic state measurements of lanes collected by the installed camera sensors. The statistics of the datasets are summarized in Table 1. We take the data from the first 60% of the entire time slot as the training set, the following 20% for validation, and the remaining 20% as the test set. For both datasets, we set both historical and predicted time window lengths  $\mathcal{T}$  and  $\tau$  to one hour. Please refer to Section B.1 for more description and analysis of the datasets.

**Implementation Details.** All experiments are performed on a Linux server with 20-core Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz and NVIDIA Tesla V100 GPU. We calculate spherical distance as the geographical distance and choose distance threshold  $\epsilon = 1\text{km}$  and prediction step size  $\xi = 12$ . We adopt three layers MLPs for asynchronous graph convolution, semi-autoregressive predictor, and meta-filters. The dimension for time encoding is set to  $d_\phi = 16$ , and dimensions for convolution filters, state evolution unit, and hidden layers of the above MLPs are all set to 64. To reduce parameter magnitude, in the implementation, we individualize the last layer's parameters but share the other parameters of MLP for meta-filters. We employ Adam optimizer to train our model, set learning rate to 0.001. ASEER and all learnable baselines are trained with an early stop criterion if the loss doesn't decrease lower on the validation set over 10 epochs.

**Evaluation Metrics.** We define six metrics to comprehensively evaluate the forecasting performance of algorithms, including C-MAE, C-RMSE, and C-MAPE to evaluate the accuracy of predicted traffic signal cycle lengths, and F-MAE, F-RMSE, and F-AAE for the traffic flow prediction evaluation. Note lower is better for all these metrics, and please refer to Section B.2 for more details of our evaluation metrics.

**Baselines.** We compare our approach with the following eleven baselines, including two heuristic approaches (LAST, HA), two classical sequence modeling approaches (TCN [2], GRU [7]), four competitive irregular time series modeling approaches (T-LSTM [3], GRU-D [4], mTAND [33], Warpformer [46]), and three competitive classical traffic forecasting approaches (DCRNN [21], GWNet [42], PDFormer [14], STAEformer [24]). For fair comparison, all learnable baseline models are set to predict the cycle lengths and unit time traffic flows by optimizing the hybrid loss function in Eq. (19) like ASEER. In addition, except for autoregressive models (*i.e.*, GRU, GRU-D, T-LSTM, DCRNN), other baselines predict in a semi-autoregressive manner with the same prediction step size as ASEER. We carefully

tune major hyper-parameters of each baseline based on their recommended settings for better performance on our datasets. Please refer to Section B.3 for more details of baselines.

### 4.2 Overall Performance

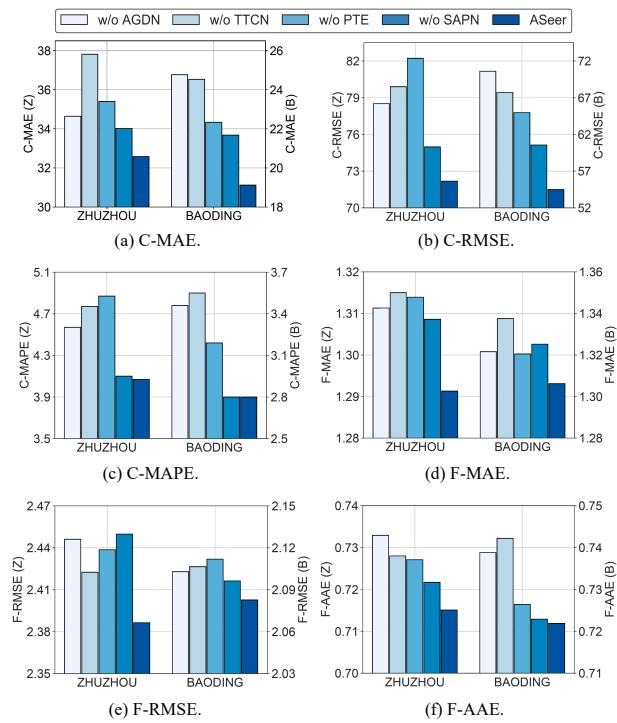
Table 2 reports the overall performance of ASEER and all compared baselines on two datasets *w.r.t.* six metrics. As can be seen, ASEER achieves the best overall performance among all the compared approaches on two datasets, which demonstrates our model's superiority in irregular traffic forecasting task. Besides, we have several observations. Firstly, all learnable approaches outperform the statistical approaches (*i.e.*, LAST, HA), which validates that the data-driven approaches to learn complex non-linear interactions within traffic data is helpful for this task. Secondly, we find CNN-based baselines TCN and GWNet do not achieve a desired performance for the reason that classical CNN applies the same parameterized filters to process sequences with different time intervals, which is incompetent to model the temporal dependency in irregular sequences. Thirdly, we observe ASEER obtains a superior overall performance than approaches (*i.e.*, GRU-D, T-LSTM, mTAND, and Warpformer) for irregular time series, as these approaches fail to model the complex spatial dependencies between large-scale sensors. From these approaches, we notice mTAND has a slight advantage in C-MAPE than ASEER on ZHUZHOU. This is probably because mTAND as a powerful approach for interpolation task performs well in the short-term future cycles' beginning times prediction. However, ASEER significantly outperforms mTAND in the other metrics. Lastly, we observe a notable performance improvement by comparing ASEER with the state-of-the-art approaches (*i.e.*, DCRNN, GWNet, and PDFormer) for classical traffic forecasting. The improvement can primarily be attributed to the capability of ASEER to effectively model asynchronous spatial dependency and irregular temporal dependency in the irregular traffic forecasting problem.

### 4.3 Ablation Study

We evaluate the performance of ASEER and its four variants on both ZHUZHOU and BAODING in all six metrics. (1) **w/o AGDN** removes the AGDN module; (2) **w/o TTCN** replaces TTCN with a 1D CNN, whose filter size is set to the maximal sequence length in the dataset; (3) **w/o PTE** removes personalized time encoding; (4) **w/o SAPN** replaces SAPN with an autoregressive MLP predictor. The results of ablation study are shown in Figure 3, As can be seen, removing any component causes notable overall performance degradation compared to ASEER, which demonstrates the effectiveness of each component. From these results, we observe **w/o TTCN** almost results in significant performance descent for all metrics on both datasets, which verifies the effectiveness of TTCN to improve classical CNN to model the temporal dependency within irregular traffic sequences. In addition, **w/o AGDN** causes a remarkable accuracy decline for all the metrics *w.r.t.* traffic flow, which validates the effect of AGDN on modeling asynchronous spatial dependency of traffic dynamics. We also observe **w/o AGDN** causes a more obvious accuracy decline on BAODING than ZHUZHOU for three metrics *w.r.t.* cycle lengths. This is probably because the distribution of cycle lengths in BAODING is denser, AGDN's smoothness induces a more precise prediction. Moreover, we notice that **w/o PTE** leads to a consistent performance reduction for all metrics on

**Table 2: Overall performance evaluated by C-MAE, C-RMSE, C-MAPE, F-MAE, F-RMSE, and F-AAE on ZHUZHOU and BAODING. The best-performing results are highlighted in bold, and the second-best results are highlighted by underline.**

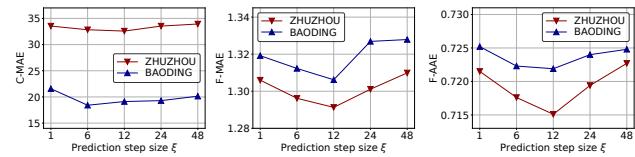
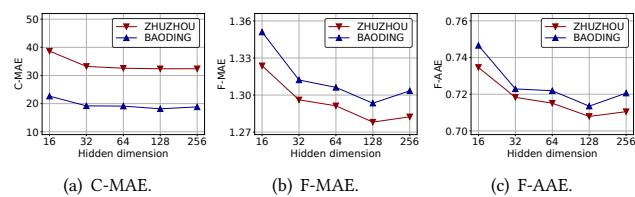
Algorithm	ZHUZHOU						BAODING					
	C-MAE	C-RMSE	C-MAPE	F-MAE	F-RMSE	F-AAE	C-MAE	C-RMSE	C-MAPE	F-MAE	F-RMSE	F-AAE
LAST	50.5386	135.2616	5.54%	1.6669	3.0995	0.9192	42.8037	106.6547	4.79%	1.7521	2.8031	0.9557
HA	52.1532	135.3569	5.76%	1.4502	2.6567	0.7998	49.7496	114.8265	5.53%	1.5449	2.4594	0.8427
TCN	43.7838	110.1670	5.01%	1.3950	2.5824	0.7818	35.8318	95.7333	4.18%	1.3815	2.2060	0.7635
GRU	40.6209	99.8693	4.82%	1.3623	2.5553	0.7524	30.4621	83.4349	3.82%	1.3576	2.1655	0.7423
T-LSTM	39.1882	87.3458	5.38%	1.3641	2.5494	0.7539	29.0845	82.5219	3.76%	1.3673	2.1887	0.7475
GRU-D	37.8531	<u>84.6255</u>	5.23%	1.3486	2.5333	0.7449	28.9117	82.5226	3.67%	1.3611	2.1735	0.7456
mTAND	37.5762	86.3045	<b>3.93%</b>	1.3563	2.5282	0.7498	27.2703	78.1066	<u>2.86%</u>	1.3575	2.1641	0.7487
Warpformer	<u>35.7369</u>	85.3125	5.02%	1.3399	2.5662	0.7405	27.8527	78.7168	3.89%	1.3554	2.1720	0.7427
DCRNN	38.5976	90.3190	4.36%	1.3318	<u>2.4438</u>	0.7348	31.0564	76.3693	3.86%	1.3681	2.1601	0.7467
GWNet	38.9913	106.6415	4.52%	1.3834	2.7915	0.7618	26.4988	84.3211	3.05%	1.3925	2.2482	0.7903
PDFormer	36.4779	86.1173	4.66%	<u>1.3170</u>	2.4575	0.7269	27.1969	<u>68.5613</u>	3.82%	<u>1.3540</u>	2.1496	0.7413
ASEER	<b>32.5803</b>	<b>72.1835</b>	<u>4.10%</u>	<b>1.2913</b>	<b>2.3864</b>	<b>0.7151</b>	<b>19.1188</b>	<b>54.4919</b>	<b>2.80%</b>	<b>1.3062</b>	<b>2.0827</b>	<b>0.7219</b>

**Figure 3: Results of ablation study. "Z" and "B" denote ZHUZHOU and BAODING, respectively.**

both datasets, which demonstrates that a well-learned personalized time encoding function to embed continuous time for each sensor can facilitate the prediction of both cycle lengths and traffic flows. Finally, by comparing ASEER with **w/o SAPN**, we observe a more obvious performance degradation on BAODING for metrics *w.r.t.* cycle lengths, which is probably because the sequence is longer on BAODING, an autoregressive model causes a severe error accumulation problem on cycle length prediction. **w/o SAPN** also shows a consistent performance descent for three metrics *w.r.t.* traffic flow, which confirms that SAPN indeed improves the long cycle length and traffic flow sequence prediction performance.

#### 4.4 Parameter Sensitivity

We conduct experiments for two important hyper-parameters, *i.e.*, the prediction step size  $\xi$  and dimension of all hidden layers, on

**Figure 4: Effect of different prediction step sizes.****Figure 5: Effect of different hidden dimensions.**

both ZHUZHOU and BAODING to study the sensitivity of these hyper-parameters. We report experimental results on metrics C-MAE, F-MAE, and F-AAE to evaluate the model's prediction performance on both cycle lengths and traffic flows.

Figure 4 shows the results of varying the prediction step size  $\xi$  from 1 to 48. As can be seen, there is a notable overall prediction performance improvement by increasing  $\xi$  from 1 (autoregressive model) to 12 (semi-autoregressive model), which demonstrates the effectiveness of SAPN to mitigate error accumulation problem in the autoregressive prediction model. However, we also observe a performance degradation when the prediction step size is too large. This is probably because a too-large prediction step size may result in under-training for SAPN to make predictions based on different elapsed times.

We vary the dimension of model's all hidden layers from 16 to 256. The results are shown in Figure 5. We can observe a remarkable prediction performance improvement by increasing the hidden dimension from 16 to 32, and the performance is continuously improving and achieves the best when the dimension is set to 128. However, a larger hidden dimension also takes more expensive computational overhead. Thus, we have to balance the performance and computation cost for the selection of model's hidden dimension.

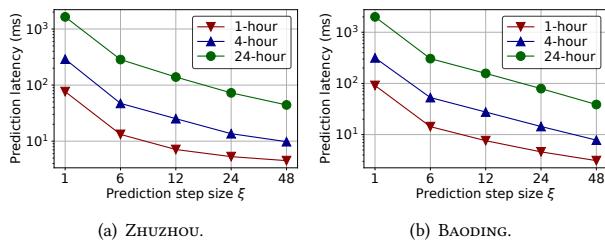


Figure 6: Prediction latency of SAPN with different prediction step sizes.

## 4.5 Prediction Efficiency Analysis

We conduct experiments to test the prediction efficiency of different models. To ensure a fair comparison, we eliminate the influence of different models on the prediction lengths by standardizing the prediction process. This involves allowing all models to predict the maximum lengths of the corresponding ground truth sequences. We analyze the prediction efficiency of SAPN in this section. Please refer to Section B.4 for more analysis on the efficiency of TTCN and the efficiency comparison between algorithms, and a comprehensive complexity analysis of ASEER can be found in Section A. **Efficiency of SAPN.** To evaluate the effect of SAPN on prediction efficiency, we conduct experiments on both ZHUZHOU and BAODING to specifically test SAPN’s average prediction latency based on different prediction step sizes  $\xi$  from 1 to 48. We report the respective results of predicting future 1, 4, and 24 hours traffic states in Figure 6. As can be seen, the prediction latency is notably reduced by comparing semi-autoregressive models ( $\xi > 1$ ) with autoregressive model ( $\xi = 1$ ) due to the reduction of total prediction steps. The magnitude of latency reduction even approaches the prediction step size when we predict longer sequences or the step size is not too large, which demonstrates the significant effectiveness of SAPN to improve prediction efficiency. We also observe with the prediction step size increasing, the prediction latency is consistently reduced, and with the predicted hours rising, the model can have a significantly higher prediction efficiency by setting a larger prediction step size. This observation indicates that we can choose a larger prediction step size with the predicted sequence length increasing for higher prediction efficiency.

## 5 RELATED WORK

**Traffic Forecasting.** Recently years, deep learning models have dominated the traffic forecasting tasks for their extraordinary capability in modeling the complex spatio-temporal characteristics of traffic data [1, 9, 10, 14, 18, 21, 32, 42, 43, 45, 47, 50]. In spatial modeling, a part of studies [43, 47] first partition a city into a grid-based region map, then utilize Convolutional Neural Networks (CNNs) to capture spatial dependencies between adjacent regions. After that, Graph Neural Networks (GNNs) [8, 16, 36, 41] are widely used to model the non-euclidean spatial dependencies in traffic data [13, 15, 23, 32]. For example, studies [21, 45] employ GNNs to model the traffic flow diffusion process in the road network. Studies [10, 11, 14, 48, 50] incorporate attention mechanism into GNNs to learn the dynamic spatial dependencies between the road network sensors. In addition to the pre-defined relational graph

derived from road networks, some works [1, 18, 31, 42, 44] attempt to directly learn the latent graph structure from traffic data. In temporal modeling, CNNs [10, 18, 42, 45, 47] and Recurrent Neural Networks (RNNs) [1, 21, 43] are frequently adopted to capture temporal dependencies within traffic data. Compared to RNNs in temporal modeling, CNNs enable parallel computing for all time steps, which exhibits extreme advantages in computational efficiency. However, all the methods of above studies are designed for the time-aligned traffic data with fixed time interval, which fails to handle the challenges of asynchronous spatial dependency and irregular temporal dependency in the irregular traffic forecasting problem.

**Irregularly Sampled Time Series.** This work is also related to the literature about learning from irregularly sampled time series, which is a kind of time series data characterized by varying time intervals between temporally adjacent observations [49]. A straightforward approach is to divide the irregularly sampled time series into a regular one with fixed time intervals [22]. However, such a temporal discretization method may lead to information loss and data missing problems [33, 34]. Recent studies tend to directly learn from irregularly sampled time series. Specifically, some studies improve RNNs by using a time gate [27], a time decay term [4], or memory decomposition mechanism [3] to adjust RNNs’ memory update for adapting irregular time series. Another line of studies introduce Neural Ordinary Differential Equations (NODEs) [6] to model the continuous dynamics in time series, and assume the latent states of time series are continuously evolving through continuous time [29, 30]. However, it is unreasonable to directly apply NODEs to model the continuous dynamics of traffic flow since it is defined over a period of time instead of a time point. Besides, attention mechanism is also applied to model irregularly sampled time series [12, 33, 40, 46, 49]. For example, Shukla and Marlin [33] proposes a multi-time attention network to learn embedding of continuous time. Zhang et al. [46] employs a doubly self-attention to learn representation from the input data unified by a warping module. Zhang et al. [49] introduce a GNN to capture time-varying dependencies between sensors by performing the graph convolution operation at all timestamps when there is an observation at an arbitrary sensor. However, it will be extremely time-consuming once the data is significantly asynchronous among large-scale sensors like us. Furthermore, the above studies primarily focus on solving irregular time series classification instead of forecasting tasks. Finally, to our knowledge, there are no prior studies attempting to modify CNNs to adapt to the irregular time series modeling.

## 6 CONCLUSION

In this paper, we investigated a new irregular traffic forecasting problem that aims to predict irregular traffic time series resulting from adaptive traffic signal controls, and presented an Asynchronous Spatio-Temporal Graph Convolutional Network, ASEER, to address this problem. Specifically, by representing the traffic sensors as nodes and linking them via a traffic diffusion graph, we first proposed an Asynchronous Graph Diffusion Network to model the spatial dependency between the time-misaligned traffic state measurements of nodes. After that, to capture the temporal dependency within irregular traffic state sequences, we devised a personalized time encoding to embed the continuous time for each

node and proposed a Transformable Time-aware Convolution Network to perform efficient temporal convolution on irregular traffic sequences. Furthermore, a Semi-Autoregressive Prediction Network was designed to iteratively predict variable-length traffic state sequences effectively and efficiently. Finally, extensive experiments on two new real-world datasets demonstrated the effectiveness of ASEER compared with eleven competitive baseline approaches in six metrics.

## REFERENCES

- [1] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems* 33 (2020), 17804–17815.
- [2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).
- [3] Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. 2017. Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 65–74.
- [4] Z Che, S Purushotham, K Cho, D Sontag, and Y Liu. 2018. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific reports* (2018), 6085–6085.
- [5] Jiming Chen, Weixin Lin, Zidong Yang, Jianyuan Li, and Peng Cheng. 2019. Adaptive ramp metering control for urban freeway using large-scale data. *IEEE Transactions on Vehicular Technology* 68, 10 (2019), 9507–9518.
- [6] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 6572–6583.
- [7] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop on Deep Learning*.
- [8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* (2016), 3844–3852.
- [9] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 364–373.
- [10] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*. 922–929.
- [11] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. 2021. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5415–5428.
- [12] Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. 2020. Set functions for time series. In *International Conference on Machine Learning*. 4353–4363.
- [13] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. 2023. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 4356–4364.
- [14] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. PDFomer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction. In *Proceedings of the AAAI conference on artificial intelligence*. 4365–4373.
- [15] Weiwei Jiang and Jiayun Luo. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications* (2022), 117921.
- [16] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations, ICLR*.
- [17] Serjeet Kohli, Steer Davies Gleave, and Luis Willumsen. 2016. Traffic Forecasting and Autonomous Vehicles. In *2016 European Transprot Conference, Barcelona*. 5–7.
- [18] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International Conference on Machine Learning*. 11906–11917.
- [19] Ibai Lana, Javier Del Ser, Manuel Velez, and Eleni I Vlahogianni. 2018. Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine* 10, 2 (2018), 93–109.
- [20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* (2015), 436–444.
- [21] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.
- [22] Zachary C Lipton, David Kale, and Randall Wetzel. 2016. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine learning for healthcare conference*. 253–270.
- [23] Fan Liu, Weijia Zhang, and Haixiu Liu. 2023. Robust Spatiotemporal Traffic Forecasting with Reinforced Dynamic Adversarial Training. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1417–1428.
- [24] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quan-jun Chen, and Xuan Song. 2023. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 4125–4129.
- [25] Ivan Marisca, Andrea Cini, and Cesare Alippi. 2022. Learning to Reconstruct Missing Data from Spatiotemporal Graphs with Sparse Observations. In *Advances in Neural Information Processing Systems*.
- [26] Fernando Moreno-Pino, Pablo M Olmos, and Antonio Artés-Rodríguez. 2023. Deep autoregressive models with spectral attention. *Pattern Recognition* (2023), 109014.
- [27] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. 2016. Phased LSTM: accelerating recurrent network training for long or event-based sequences. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 3889–3897.
- [28] Rezaur Rahman and Samiul Hasan. 2023. A deep learning approach for network-wide dynamic traffic prediction during hurricane evacuation. *Transportation Research Part C: Emerging Technologies* 152 (2023), 104126.
- [29] Yulia Rubanova, Ricky TQ Chen, and David Duvenaud. 2019. Latent ODEs for irregularly-sampled time series. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 5320–5330.
- [30] Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. 2022. Modeling irregular time series with continuous recurrent units. In *International Conference on Machine Learning*. 19388–19405.
- [31] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1567–1577.
- [32] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. 2022. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment* (2022), 2733–2746.
- [33] Satya Narayan Shukla and Benjamin Marlin. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *International Conference on Learning Representations*.
- [34] Satya Narayan Shukla and Benjamin M Marlin. 2020. A survey on principles, models and methods for learning from irregularly sampled time series. *arXiv preprint arXiv:2012.00168* (2020).
- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems* 30 (2017).
- [36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.
- [37] Chen Wang, Bertrand David, René Chalon, and Chuantaoy Yin. 2016. Dynamic road lane management study: A Smart City application. *Transportation research part E: logistics and transportation review* 89 (2016), 272–287.
- [38] Yizhe Wang, Xiaoguang Yang, Hailun Liang, Yangdong Liu, et al. 2018. A review of the self-adaptive traffic signal control system based on future traffic environment. *Journal of Advanced Transportation* 2018 (2018).
- [39] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2019. A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117* (2019).
- [40] Yuxi Wei, Junlong Peng, Tong He, Chenxin Xu, Jian Zhang, Shirui Pan, and Siheng Chen. 2023. Compatible transformer for irregularly sampled multivariate time series. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1409–1414.
- [41] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* (2020), 4–24.
- [42] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1907–1913.
- [43] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Zhenhui Li, Jieping Ye, and Didi Chuxing. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*. 2588–2595.
- [44] Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. 2023. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in Neural Information Processing Systems* 36 (2023).

- [45] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3634–3640.

[46] Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. 2023. Warpformer: A Multi-Scale Modeling Approach for Irregular Clinical Time Series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3273–3285.

[47] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*. 1655–1661.

[48] Weijia Zhang, Hao Liu, Yanchi Liu, Jingbo Zhou, and Hui Xiong. 2020. Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1186–1193.

[49] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Graph-Guided Network for Irregularly Sampled Multivariate Time Series. In *International Conference on Learning Representations*.

[50] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*. 1234–1241.

## A COMPLEXITY ANALYSIS

In this section, we analyze the time complexity of ASEER. Given the historical and predicted time windows, we let  $T^i, M^i, N_i$  denote the number of historical traffic state measurements, total prediction steps, and neighbors of node  $v^i$ , denote  $N$  as the number of nodes,  $\xi$  as prediction step size, and let  $d$  denote the dimensions for all feature representation vectors to ease the presentation.

The time complexity of AGDN module is mainly determined by processing all the traffic messages stored in the buffers. Each node diffuses its traffic state measurements to its neighbors, and the nodes would clear their buffers after performing asynchronous graph convolution on the stored traffic messages (*i.e.*, a message would be computed only once), thus each node  $v^i$  would lead to  $T^i N_i$  messages for computation. Therefore, the time complexity of AGDN module would be  $O(\sum_{i=1}^N T^i N_i d^2)$ .

For the TTCN module, the time complexity is mainly from generating time-aware convolution filters, where a  $T^i \times d$  dimensional filter incurs a cost of  $O(T^i d^2)$ . We generate  $d$  filters, thus we obtain the time complexity for TTCN module as  $O(\sum_{i=1}^N T^i d^3)$ .

For the SAPN module, as each prediction step incurs a cost of  $O(d^2 + d\xi)$ , the time complexity of  $M^i$  prediction steps for all nodes would be  $O(\sum_{i=1}^N M^i (d^2 + d\xi))$ .

The time complexity of generating personalized time encoding for a continuous time is  $O(d)$ , which is a lower-order term when considering the time complexity of each module, and thus can be neglected. Hence, we deduce the overall time complexity of ASEER:

$$O\left(\sum_{i=1}^N \left(T^i d^2 (N_i + d) + M^i (d^2 + d\xi)\right)\right). \quad (20)$$

Fortunately, the time complexity can be further reduced in practical real-time traffic forecasting applications. This is attributed to our model's character that the spatial representations acquired from AGDN can be asynchronously computed once the corresponding historical traffic measurements are observed. Assuming a traffic forecasting request is occurring at timestamp  $t$ , almost all spatial representations during the historical time window  $\mathcal{T}$  have been obtained. We only need to re-compute the spatial representations  $\bar{h}_1^i$  and  $\bar{h}_{T^i}^i$  for the first chronological measurement and the remaining messages, inducing a time complexity  $O(\sum_{i=1}^N (|\mathcal{B}_1^i| + |\mathcal{B}_{T^i}^i|) d^2)$ , where  $|\mathcal{B}_1^i|$  and  $|\mathcal{B}_{T^i}^i|$  denotes the number of stored messages for the computation of  $\bar{h}_1^i$  and  $\bar{h}_{T^i}^i$ . Moreover, if we restrict to selecting a maximum of  $K$  messages (*e.g.*, the latest  $K$  messages) from these two buffers for computation, the time complexity can be further optimized to  $O(\sum_{i=1}^N (\text{Min}(|\mathcal{B}_1^i|, K) + \text{Min}(|\mathcal{B}_{T^i}^i|, K)) d^2)$ . Consequently, the practical time complexity of ASEER is reduced to:

$$O\left(\sum_{i=1}^N \left(\left(\text{Min}(|\mathcal{B}_1^i|, K) + \text{Min}(|\mathcal{B}_{T^i}^i|, K)\right) d^2 + T^i d^3 + M^i (d^2 + d\xi)\right)\right). \quad (21)$$

We verify the efficiency of ASEER and its components in Section 4.5 and Section B.4.

## B SUPPLEMENTARY EXPERIMENTS

### B.1 Data Description and Analysis

**B.1.1 Datasets Description.** The statistics of the datasets are summarized in Table 1. Specifically, there are total 19,824,504 and 13,093,975 traffic state measurements on ZHUZHOU and BAODING, and the missing period ratios of the two datasets are 44.2% and 27.2%, respectively. Each measurement includes information about the beginning and end timestamps and cycle length of the traffic signal cycle, as well as the lane's traffic flow, *i.e.*, the number of vehicles passing through the camera of lane, during the signal cycle. Besides, ZHUZHOU has 620 lanes with sensors and ranges from July 20, 2022 to October 2, 2022. BAODING has 264 lanes with sensors and ranges from December 1, 2021 to February 25, 2022. The average, maximal ground truth sequence length to be predicted for the future one hour is 57, 213 on ZHUZHOU, and 64, 155 on BAODING.

**B.1.2 Datasets Analysis.** The overall distributions of traffic signal cycle lengths on two datasets are depicted in Figure 7, where we can observe the cycle lengths can significantly vary from around 40 to 200 seconds on both datasets, indicating the pronounced irregularity within time series, and BAODING has a denser cycle length distribution than ZHUZHOU.

Besides, Figure 8 illustrates temporal distributions of traffic signal cycle lengths and traffic flows across different hours on both datasets. We can observe cycle length and traffic flow consistently exhibit higher values during the daytime periods compared to overnight periods. Moreover, they display similar peak patterns during the morning and evening rush hours and tend to vary in a positively correlated manner.

To further investigate the correlations between these two traffic states, we illustrate the variations in traffic flow distributions across different cycle lengths and vice versa in Figure 9. As can be seen in Figure 9(a) and Figure 9(b), traffic flow maintains an upward trend at first along with the increase of cycle length. A similar positive correlation can also be observed in Figure 9(c) and Figure 9(d), which display the variations in cycle length distributions across distinct traffic flows. However, we notice that with a further increase in cycle length, traffic flow tends to decrease. A similar situation is also shown in Figure 9(c). This can be attributed to the fact that although a positive correlation is expected between traffic flow and cycle length for the same lane, the lanes with the longest cycle lengths may not necessarily correspond to the highest traffic flows due

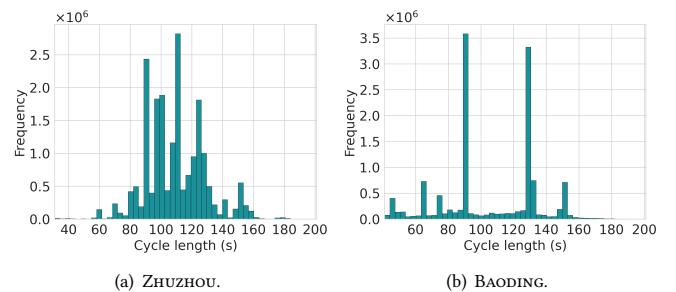


Figure 7: Overall distributions of traffic signal cycle lengths.

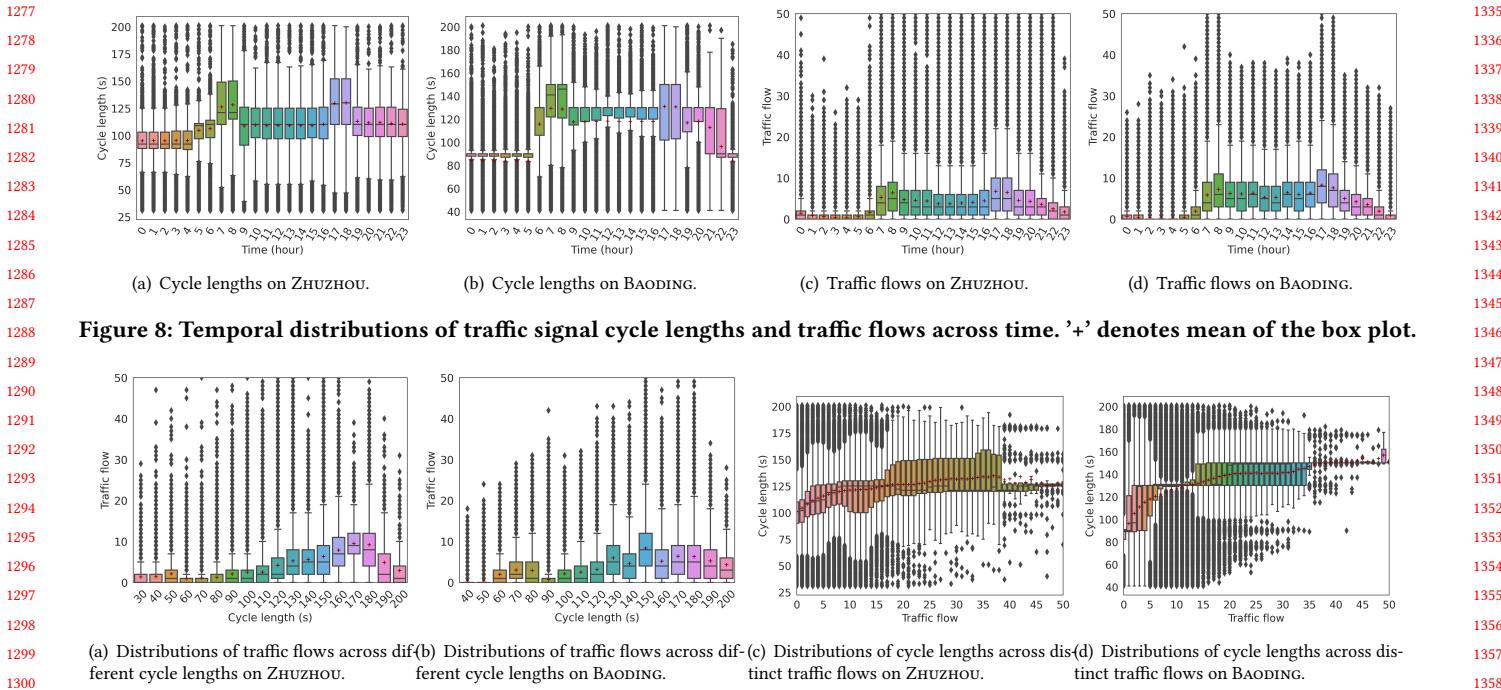


Figure 8: Temporal distributions of traffic signal cycle lengths and traffic flows across time. '+' denotes mean of the box plot.

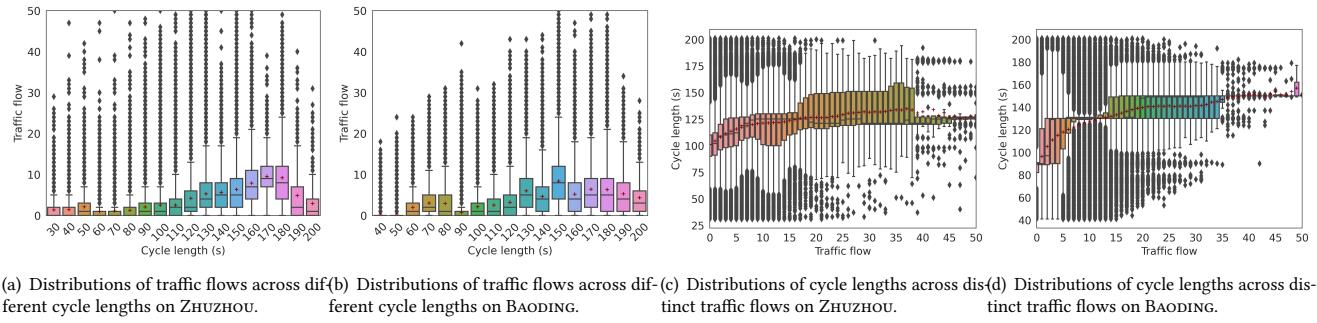


Figure 9: Correlations between traffic flow and traffic signal cycle length. '+' denotes mean of the box plot.



Figure 10: Spatial distributions of camera sensors and corresponding average traffic flows and cycle lengths on Zhuzhou.

to different traffic conditions and signal control strategies among these lanes, and vice versa.

Additionally, Figure 10 displays the spatial distributions of camera sensors and corresponding average traffic flows and cycle lengths on ZHUZHOU as a representative. It can be noticed that both traffic flows and cycle lengths exhibit remarked geographical proximity, indicating that neighboring sensors tend to have similar traffic states. This finding provides partial justification for the effectiveness of the spatial dependency modeling component, AGDN, in the irregular traffic forecasting task.

## B.2 Evaluation Metrics

We define six metrics to comprehensively evaluate the forecasting performance of algorithms. Lower is better for all six metrics.

Specifically, to evaluate the accuracy of predicted traffic signal cycle lengths, we quantify the predicted errors of both the beginning timestamps and cycle lengths via masked Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute

Table 3: Hyper-parameters of ASEE.

Hyper-parameter	ZHUZHOU	BAODING
Distance threshold $\epsilon$	1km	1km
Prediction step size $\xi$	12	12
Hidden dimension	64	64
Time encoding dimension $d_\phi$	16	16
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Early stop patience	10	10

Percentage Error (MAPE):

$$\begin{aligned}
 \text{C-MAE} &= \frac{1}{2 \times N \times K^i_{\text{I}}} \sum_{i=1}^N \sum_{k=1}^{K^i} \left( |\hat{b}_k^i - b_k^i| + |\hat{p}_k^i - p_k^i| \right) \times \zeta_k^i, \\
 \text{C-RMSE} &= \sqrt{\frac{1}{2 \times N \times K^i_{\text{I}}} \sum_{i=1}^N \sum_{k=1}^{K^i} \left( (\hat{b}_k^i - b_k^i)^2 + (\hat{p}_k^i - p_k^i)^2 \right) \times \zeta_k^i}, \\
 \text{C-MAPE} &= \frac{100\%}{2 \times N \times K^i_{\text{I}}} \sum_{i=1}^N \sum_{k=1}^{K^i} \left( \frac{|\hat{b}_k^i - b_k^i|}{\delta_k^i} + \frac{|\hat{p}_k^i - p_k^i|}{p_k^i} \right) \times \zeta_k^i.
 \end{aligned} \tag{22}$$

where  $K^i$  denotes the number of ground truth traffic states of each sensor for evaluation,  $K_1^i$  is the number of observed measurements.

Since traffic flows depend on the corresponding traffic signal cycles, it's incomparable between the predicted and ground truth traffic flows if they are misaligned in signal cycles. Thus, we introduce two types of metrics to evaluate the prediction accuracy of traffic flows from multiple aspects. First, we assume all the traffic signal cycle lengths can be accurately predicted and use the ground truth cycle lengths for calculation, then we can directly compare the predicted and ground truth traffic flows via the following masked MAE and RMSE metrics:

$$\begin{aligned} \text{F-MAE} &= \frac{1}{N \times K_1^i} \sum_{i=1}^N \sum_{k=1}^{K^i} |\hat{f}_k^i - f_k^i| \times \zeta_k^i, \\ \text{F-RMSE} &= \sqrt{\frac{1}{N \times K_1^i} \sum_{i=1}^N \sum_{k=1}^{K^i} (\hat{f}_k^i - f_k^i)^2} \times \zeta_k^i. \end{aligned} \quad (23)$$

Second, without the above assumption for cycle lengths, by computing traffic flow density at any timestamp, we calculate the masked Accumulative Absolute Error (AAE) between predicted and ground truth traffic flow density at identical timestamps:

$$\text{F-AAE} = \frac{1}{Z} \sum_{i=1}^N \sum_t |\hat{\rho}_t^i - \rho_t^i| \times \eta_t^i, \quad (24)$$

where  $\hat{\rho}_t^i = \hat{f}_k^i / \hat{p}_k^i$ ,  $t \in [\hat{b}_k^i, \hat{t}_k^i]$  and  $\rho_t^i = f_k^i / p_k^i$ ,  $t \in [b_k^i, t_k^i]$  are the predicted and ground truth traffic flow densities of sensor  $v^i$  at timestamp  $t$ , respectively.  $\eta_t^i$  is the mask term at  $t$ , which equals one if  $\rho_t^i$  can be obtained from observed measurement, and zero otherwise. In our experiments, the timestamp is in seconds, and we use a normalization term  $Z$  to obtain the average result in minutes.

### B.3 Baselines

We compare our approach with the following eleven baselines. These baseline models take the same inputs as our ASEER by directly utilizing observed traffic state measurements. Moreover, we pad the input sequences to the median of all sequence lengths if models necessitate a fixed-length input. All these models aim to predict both traffic flow and cycle lengths by optimizing the hybrid loss function in equation Eq. (19). The autoregressive models, i.e., GRU, GRU-D, T-LSTM and DCRNN, iteratively predict the next step traffic states based on their previous predictions. Since the other non-autoregressive models require the predicted sequence length to be fixed, to enable variable-length sequence prediction, we allow them to predict in a semi-autoregressive way that they iteratively predict a fixed-length sub-sequence based on observed and previously predicted sequences. The prediction step size is set the same as ours to ensure a fair comparison. We carefully tuned major hyperparameters of each baseline based on their recommended settings for better performance on our datasets.

- **LAST** predicts future traffic states using each sensor's last historical traffic state measurement.
- **HA** predicts future traffic states using the average of each sensor's historical traffic state measurements.
- **TCN** [2] is the temporal convolutional network consisting of causal and dilated convolutions. We apply it to our

datasets by padding or intercepting all the sequences to a fixed length. We stack 6 temporal convolution layers with filter size of 3.

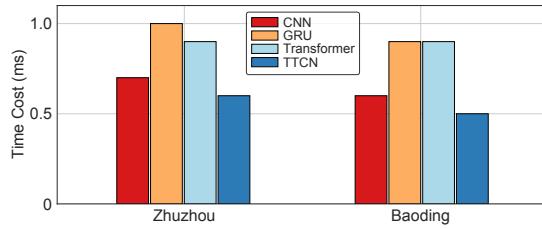
- **GRU** [7] is a powerful variant of recurrent neural networks with a gated recurrent unit.
- **T-LSTM** [3] is a time-aware Long-Short Term Memory (LSTM) model with memory decomposition for irregular time series classification. We modify it to predict traffic states using a LSTM-based decoder.
- **GRU-D** [4] improves GRU with a time-aware decay mechanism for irregular time series classification. We modify it to predict traffic states using a GRU-based decoder.
- **mTAND** [33] is a state-of-the-art transformer-based approach for irregularly sampled multivariate time series classification and interpolation tasks. It adopts multi-time attention with time embedding to produce a fixed-length representation of a variable-length time series. The reference point number is set to 64.
- **Warpformer** [46] is another state-of-the-art transformer-based approach for irregularly sampled multivariate time series classification tasks. It employs a warping module to unify inputs and a doubly self-attention module for representation learning. We set the lengths of warp layers to 0, 24.
- **DCRNN** [21] is a representative approach based on GNNs and RNNs for classical traffic forecasting tasks, which replaces the matrix multiplications in GRU with a graph convolution operation. The used graph structure is the same as ASEER, and the diffusion step is set to 1. To apply DCRNN to our problem, we pad the input traffic sequences of all nodes to the same length.
- **GWNet** [42] is a representative approach based on GNNs and CNNs for classical traffic forecasting. It stacks multiple spatial-temporal blocks that are constructed by the graph convolution layer and gated TCN layer, where the graph convolution is performed on the combination of pre-defined and self-learned adjacency matrix. The pre-defined graph structure is the same as ASEER. We stack 3 blocks with 4 convolution layers and set the convolution filter size to 3. It adopts the same padding strategy as DCRNN.
- **PDFormer** [14] is a state-of-the-art approach based on transformer for classical traffic forecasting. It adopts self-attentions for both spatial and temporal dependencies modeling. A graph-masked self-attention mechanism is employed to capture both geographic and semantic spatial dependencies and a delay-aware feature transformation module is used to model the time delay in spatial information propagation. The depth of encoder layers is set to 2. It adopts the same padding strategy as DCRNN.

### B.4 Prediction Efficiency Analysis

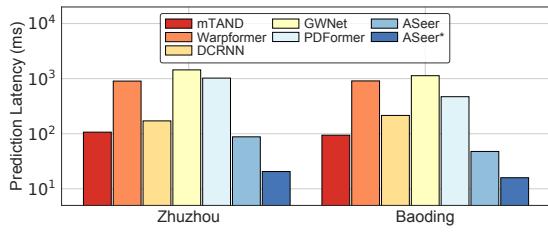
We conduct experiments to test the prediction efficiency of different models. To ensure a fair comparison, we eliminate the influence of different models on the prediction lengths by standardizing the prediction process. This involves allowing all models to predict the maximum lengths of the corresponding ground truth sequences.

1509 **Table 4: Variation of prediction performance and latency by restricting messages usage in the buffer. ASEER\* denotes ASEER with**  
 1510 **a restriction  $K = 10$  in Eq. (21).**

Dataset	Algorithm	C-MAE	C-RMSE	C-MAPE	F-MAE	F-RMSE	F-AAE	Latency
ZHUZHOU	ASEER	32.5803	72.1835	4.10%	1.2913	2.3864	0.7151	87.9ms
	ASEER*	32.8299	72.7056	4.13%	1.2939	2.3954	0.7166	20.6ms
BAODING	ASEER	19.1188	54.4919	2.80%	1.3062	2.0827	0.7219	47.7ms
	ASEER*	19.3724	55.1685	2.84%	1.3078	2.0871	0.7232	15.9ms



1518 **Figure 11: Efficiency of different modules in temporal modeling.**



1526 **Figure 12: Efficiency comparison between different models.**

1539 **Efficiency of TTCN.** To study TTCN's efficiency, we replace TTCN  
 1540 with several commonly used modules in temporal modeling, *i.e.*,  
 1541 CNN, GRU, and Transformer, and test their running time costs. As il-  
 1542 lustrated in Figure 11, TTCN achieves more than 40% and 33% faster  
 1543 results than GRU and transformer, respectively, on both datasets.  
 1544 Furthermore, to our surprise, TTCN exhibits even faster than CNN.  
 1545 This is probably because TTCN can directly handle variable-length  
 1546 sequences with transformable filter sizes, while CNN is limited to  
 1547 processing fixed-length sequences via padding or clipping, thus it  
 1548 may cost additional time to process longer sequences beyond their  
 1549 original lengths. All the observations demonstrate the efficiency of  
 1550 TTCN.

1551 **Efficiency Comparison.** Figure 12 displays the comparison of  
 1552 average prediction latency between ASEER and other state-of-the-  
 1553 art baseline models. To enable mTAND, Warpformer, GWNet, and  
 1554 PDFormer to predict the variable-length sequence, we let them per-  
 1555 form in an autoregressive way. We can observe that ASEER (87.9ms,  
 1556 47.7ms with  $\xi = 12$ ) is markedly more efficient than other state-  
 1557 of-the-art baselines on both ZHUZHOU and BAODING. Moreover, by  
 1558 restricting the usage of buffer messages  $K = 10$  in Eq. (21), ASEER\*  
 1559 can even achieve much faster prediction latency (20.6ms, 15.9ms)  
 1560 without losing too much prediction accuracy compared to ASEER  
 1561 as reported in Table 4. The results verify our model's efficiency in  
 1562 practical real-time traffic forecasting applications. The efficiency  
 1563 of ASEER is mainly attributed to three reasons: (1) SAPN is much  
 1564 more efficient than a fully autoregressive model; (2) TTCN can be  
 1565 efficiently computed in parallel like CNN; and (3) AGDN enables

1569 pre-computation of spatial representations in advance, allowing  
 1570 them to be readily available for responding to real-time traffic fore-  
 1571 casting requests.