Benchmarking Contextual Factor Generalizability in Spatiotemporal Crowd Flow Prediction

Liyue Chen, Leye Wang

Abstract—Contextual features are important data sources for building spatiotemporal crowd flow prediction (STCFP) models. However, the difficulty of applying context lies in the unknown generalizability of both contextual features (e.g., weather, holiday, and points-of-interests) and context modeling techniques across different scenarios. In this paper, we develop an experimental platform composed of large-scale spatiotemporal crowd flow data, contextual data, and state-of-the-art spatiotemporal prediction models to conduct a comprehensive experimental study to quantitatively investigate the generalizability of different contextual features and modeling techniques in three urban crowd flow prediction scenarios (bike flow, metro passenger flow, and electric vehicle charging demand). In particular, we develop a general taxonomy of context modeling techniques based on extensive investigations in prevailing research. With three real-world datasets including millions of records and rich context data, we have trained and tested hundreds of different models. Our results reveal several important observations: (1) Using more contextual features may not always result in better prediction with existing context modeling techniques; in particular, the contextual feature combination of holiday and temporal position can provide more generalizable beneficial information than other contextual feature combinations. (2) In context modeling techniques, using a gated unit to incorporate raw contextual features into the state-of-the-art prediction model has good generalizability. Besides, we also offer several suggestions about incorporating contextual factors for practitioners who want to build STCFP applications. From our findings, we call for future research efforts devoted to developing new context processing and modeling solutions to fully exploit the potential of contextual features for STCFP.

Index Terms—Spatiotemporal Prediction, Context, Generalizability

I. INTRODUCTION

Contextual factors (e.g., weather, holiday) have proven to be useful as features for a wide variety of tasks in spatiotemporal crowd flow prediction (STCFP) tasks such as bike-sharing demand prediction and metro passenger flow forecasting. In general, context is any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical/computational object. Understanding context is critical for crowd flow prediction tasks as it may provide extra useful information for learning specific spatiotemporal patterns.

In STCFP studies, researchers usually consider spatial and temporal dependencies and discover many kinds of spatiotemporal knowledge. Temporal dependency reveals the relationship between future and past, while spatial dependency mainly denotes the relationship between different geographic locations. Meanwhile, context also plays a key role. Taking weather context as an example, rising temperatures will promote the usage of bike-sharing, heavy rains will lessen the usage of both bike-sharing and online ride-hailing. Prior studies have proposed many dedicated techniques to capture spatiotemporal dependencies (e.g., attention mechanism and graph convolution network) and the generalizability of spatiotemporal knowledge has been well-studied.

Previous studies have focused on utilizing specific contextual features in certain applications, while the generalizability of contextual features is still under-investigated: for example, POIs (Points-Of-Interest) data is effective in taxi demand prediction problem, but whether POIs data is still beneficial in other scenarios remains to be evaluated. Besides, while pioneering studies propose various context modeling techniques such as Adding Embedding and Gating, it is still hard to select appropriate context modeling techniques for a given problem since the generalizability of these modeling techniques is unknown.

Overall, analyzing the generalizability of contextual features and context modeling techniques is of significant value for building effective spatiotemporal prediction models. Meanwhile, this analysis is challenging in the following aspects:

- **Unknown generalizability of contextual features.** Most studies directly consider a specific set of contextual features without carefully analyzing whether the selection of these features is optimal. To the best of our knowledge, no previous study has thoroughly compared the generalizability of contextual features across different application scenarios.
- **Lacking a taxonomy of context modeling techniques.** Previous studies propose various context modeling techniques. For the convenience of analyzing the generalizability of context modeling techniques, a comprehensive taxonomy is desired.
- **Unknown generalizability of context modeling techniques.** Though several context modeling techniques have been proposed, it is hard to determine a suitable modeling technique given an STCFP task. As far as we know, no previous study has analyzed the generalizability of different context modeling techniques across scenarios.

To sum up, in this paper, we aim to analyze the generalizability of both contextual features and context modeling techniques in STCFP applications. We conduct a large-scale analytical and experimental empirical study. Particularly, we try to give some design guidelines for the research community.
and make the following contributions:

- To the best of our knowledge, this is the first study that focuses on investigating how contextual factors (both features and modeling techniques) would generally and quantitatively impact spatiotemporal crowd flow prediction performance in various practical scenarios. By answering this critical but under-investigated problem, we expect that this research can inspire researchers and practitioners to learn how to efficiently incorporate contextual factors in STCFP models and applications.

- By surveying recent STCFP studies, we categorize the mostly-used contextual features in literature, including weather, holiday, temporal position, and POIs. For weather, we further decompose it into seven common variables, including temperature, humidity, visibility, etc. In addition, by investigating existing context modeling techniques, we develop a novel unified taxonomy for the context modeling techniques in state-of-the-art STCFP models. It is worth noting that, with this taxonomy, we summarize 14 possible context modeling techniques in total; while ten techniques are leveraged in literature, the other four are newly established and investigated by our work.

- We develop an experimental platform and conduct a comprehensive experimental study to quantitatively investigate the generalizability of different contextual features and modeling techniques across three STCFP application scenarios (bike-sharing, metro, and electronic vehicle). According to our experiment results, there is a surprising and counter-intuitive observation — with current context modeling techniques, introducing more contextual features (e.g., weather) would often degrade the prediction performance. Specifically, the contextual features of holiday and temporal position can usually provide beneficial information, while weather and POIs often fail to offer significant improvements (may even incur negative impacts). Based on the experimental observations, we offer several suggestions about incorporating contextual factors for practitioners who want to build STCFP applications. From our findings, we also call for future research efforts devoted to developing new context processing and modeling solutions to fully exploit the potential of contextual features for STCFP.

- Our platform is released with the spatiotemporal crowd flow data, contextual data, and state-of-the-art STCFP models for reproducing our experiment results. Researchers may also leverage our platform to conduct experiments on their interested STCFP scenarios for investigating the context effectiveness and generalizability.

II. ANALYTICAL STUDIES ON CONTEXTUAL FEATURES

In this section, we conduct analytical studies on contextual features. We revisit contextual features mentioned in recent STCFP studies and introduce the contextual data preprocessing methods.

A. Contextual Features

Weather. In general, weather contexts such as temperature, humidity, wind speed, and weather state (e.g., cloudy and thunderstorms) refer to short-term changes in the atmosphere, which can influence crowd flow. For instance, rising temperatures will increase the use of bike-sharing. Heavy rains and strong winds will reduce the use of bike-sharing and online ride-hailing. The crowd flow in different STCFP application scenarios may be affected by different kinds of weather contexts. The second column of Table I summarizes the weather context usage statistics in existing studies. We observe that temperature and weather state are the most widely considered contextual features.

Holiday. In reality, the crowd flow daily patterns are closely related to holidays. There is an obvious migration flow from urban residential areas to business areas during workdays, but this pattern is unclear on holidays. Hoang et al. also reveal that the subway traffic pattern on weekends is obviously different from that on weekdays.

Temporal Position. Crowd flow patterns are different for each day in a week. As an illustration, the crowd flow on Friday night is usually higher than that on Thursday night. The crowd flow hourly patterns in one day may also be different. For example, the peak traffic in the morning and evening is significantly higher than the traffic in off-peak hours. This phenomenon reveals that the crowd flow patterns at different temporal periods are dissimilar. We use two hard-coded indicators to record temporal position features of crowd flow:

- HourOfDay indicates where the current time is in one day (what time it is), and its value ranges from 0 to 23 (representing 0 o’clock to 23 o’clock). A widely adopted approach is to treat HourOfDay as discrete features and then transform them by one-hot encoding.

- DayOfWeek indicates where the current time is in one week (which day it is). The value range of DayOfWeek is between Monday and Sunday. DayOfWeek can also be transformed by one-hot encoding.

Points Of Interests. POIs are specific point locations including residential areas, business areas, and tourist attractions in a city. POIs can greatly improve our understanding of these locations’ traffic patterns, which are usually considered as an important type of information to model spatial correlations.

B. Contextual Features Preprocessing Methods

Contextual features may be either continuous or discrete. Continuous features are with specific numerical values, e.g., temperature (°F), humidity (%), and wind speed (m/s). Discrete contextual features are with labels, e.g., weather state (sunny/rainy/...) and holiday (workday/holiday).

Continuous features can be directly fed into various machine learning models such as neural networks as features. However, discrete features like weather state, holiday, temporal position, and POIs, should be encoded first by certain transformations.

2https://github.com/Liyue-Chen/STCFPCONTEXT

3https://www.ncei.noaa.gov/news/weather-vs-climate
### Table I
**Contextual features and context modeling techniques in STCFP studies.** (T: Temperature; H: Humidity; V: Visibility; WS: Wind Speed; WD: Wind Degree; AQ: Air Quality; S: Weather State)

| Context Modeling Technique | Weather | Holiday | Temporal Position | POIs | Modeling Method |
|----------------------------|---------|---------|-------------------|------|-----------------|
| **Bike-sharing**           |         |         |                   |      |                 |
| Li et al. [1]               | T;WS;S  |         |                   |      | Concat (traditional machine learning) |
| Yang et al. [12]            | T;H;V;WS;S | ✓      |                   |      | Concat (traditional machine learning) |
| Chai et al. [6]             | T;WS;S  | ✓       |                   | ✓    | Emb-Concat      |
| Li et al. [13]              | T;WS;S  | ✓       |                   | ✓    | Concat (traditional machine learning) |
| **Ridesharing**            |         |         |                   |      |                 |
| Tong et al. [9]             | T;WS;WD;AQ;S | ✓      | ✓                 | ✓    | Concat (traditional machine learning) |
| Ke et al. [14]              | T;H;V;WS;S | ✓      |                   | ✓    | LSTM-Add        |
| Wang et al. [15]            | T;AQ;S  | ✓       |                   | ✓    | MultiEmb-Concat |
| Zhu et al. [16]             |         | ✓       |                   | ✓    | Raw-Concat      |
| Yao et al. [17]             | T;S     | ✓       |                   | ✓    | EarlyConcat     |
| Saadallah et al. [18]       | T;WS;S  | ✓       |                   | ✓    | Concat (traditional machine learning) |
| **Metropolitan Crowd Flow**|         |         |                   |      |                 |
| Liu et al. [2]              | S       | ✓       |                   | ✓    | MultiEmb-Concat |
| **Traffic Flow**            |         |         |                   |      |                 |
| Yi et al. [19]              |         |         |                   | ✓    | Emb-Add         |
| Zhang et al. [20]           |         |         |                   | ✓    | EarlyConcat     |
| Barnes et al. [21]          |         |         |                   | ✓    | Concat (traditional machine learning) |
| Zheng et al. [22]           | T;WS;V;S| ✓       |                   | ✓    | Feature engineering + CNN |
| Zhang et al. [23]           | WS;T;S  | ✓       |                   | ✓    | MultiEmb-Concat |
| Yuan et al. [24]            | S       | ✓       |                   | ✓    | Emb-Concat      |
| **Citywide Crowd Flow**     |         |         |                   |      |                 |
| Hoang et al. [4]            | T       |         |                   | ✓    | Concat (traditional machine learning) |
| Zhang et al. [10]           |         |         |                   | ✓    | Raw-Add         |
| Zhang et al. [25]           | T;WS;S  | ✓       |                   | ✓    | Raw-Add         |
| Zonoozi et al. [26]         |         |         |                   | ✓    | Raw-Add         |
| Zhang et al. [11]           | T;WS;S  | ✓       |                   | ✓    | Raw-Gating      |
| Chen et al. [27]            | S       | ✓       |                   | ✓    | EarlyConcat     |
| Sun et al. [28]             | T;WS;S  | ✓       |                   | ✓    | Emb-Gating/Add  |
| Jiang et al. [29]           |         |         |                   | ✓    | EarlyConcat     |

(e.g., one-hot encoding or embedding). For example, *HourofDay* can be encoded into a one-hot vector of length 24, which represents the index of an hour in a day. *DayofWeek* can be encoded into a vector of length 7. For POIs, researchers often count the POIs density of different categories as features [9].

Note that, one-hot encoding may explode the feature dimensions which may lead to the curse of dimensionality issue. Some researchers reduce feature dimension by manual coding, e.g., categorizing the weather state into good weather (sunny, cloudy) and bad weather (rainy, storm, dusty) [4], [27]. Other researchers use *Embedding* [6], [25] to achieve dimension reduction. We will elaborate on context modeling techniques in the next section.

### III. Analytical Studies on Context Modeling Techniques

To illustrate the context modeling techniques more clearly, we first introduce a flexible and general spatiotemporal modeling framework (Fig. 1). The neural networks in our framework denote deep spatiotemporal networks, such as STMGCN [7] and STMeta [8]. The spatiotemporal neural networks take spatiotemporal features (e.g., time series and spatial graphs) as inputs, learning diverse spatiotemporal representations. Our framework classifies existing context modeling techniques into **Early Joint Modeling** and **Late Fusion**. Early joint modeling fuses contextual features with spatiotemporal features at the input stage, while late fusion fuses contextual features with spatiotemporal representations learned by spatiotemporal networks. Note that our taxonomy is developed mainly for deep learning models (see the last column in Table I). Traditional statistical learning models such as XGBoost [32] usually directly fuse contextual features by concatenating [9].

#### A. Early Joint Modeling

Early joint modeling refers to fusing contextual features with raw spatiotemporal inputs before capturing spatiotemporal dependencies via spatiotemporal modeling units. For example, Lin et al. [33] combine the features of DayofWeek, HourofDay and POIs’ population distribution map by using adding, and then apply the ResPlus structure to capture spatial dependencies between distant locations. Yao et al. [17], [34] first concatenate spatial features and contextual features (including weather and holiday), and then capture temporal patterns by LSTM (Long Short-Term Memory) [35]. In a word, *Concatenate* and *Add* both can fuse contextual features and spatiotemporal features at the early stage, and we name...
the above modeling techniques EarlyAdd and EarlyConcat, respectively. Note that $E \in \mathbb{R}^{T \times N \times P}$ and $ST \in \mathbb{R}^{T \times N \times Q}$ are the number of contextual features and raw spatiotemporal inputs. $T$ is the number of historical observations and $N$ is the number of locations. $P$ and $Q$ are the dimensions of contextual features and spatiotemporal inputs. The output of early joint modeling is $O_e$. The formula of EarlyConcat is:

$$O_e = \text{Concat}(E; ST) \in \mathbb{R}^{T \times N \times (P + Q)} \quad (1)$$

The formula of EarlyAdd is:

$$O_e = E \cdot W_e + ST \cdot W_{st} + b \quad (2)$$

where $W_e \in \mathbb{R}^{P \times D}$, $W_{st} \in \mathbb{R}^{Q \times D}$ and $b \in \mathbb{R}^D$ are trainable parameters; $D$ is the dimension of the fused features.

B. Late Fusion

Late Fusion denotes fusing contextual features and latent spatiotemporal representations learned by high-level layers of neural networks. In particular, we elaborate on these methods in two stages, (i) context representation stage and (ii) feature fusion stage. The first stage is to capture different patterns of context; the second stage is to fuse context representations with spatiotemporal latent representations for prediction.

1) Context Representation Stage:

**Raw.** No transformations are applied to contextual features. **Embedding.** The embedding technique is widely used in various fields, such as NLP (Natural Language Processing). It maps sparse high-dimension features to dense low-dimension features. Many studies use fully-connected layers for embedding \([2], [5], [15]\). For instance, Zhang et al. \([25]\) use two fully-connected layers upon contextual features, and the first layer acts as an embedding layer. **Multiple Embedding.** Different contextual features can be fed into multiple embedding layers \([2], [5], [15]\). For example, Liu et al. \([2]\) use different embedding layers to model weather and temporal position, respectively.

**LSTM.** Some contextual features (e.g., weather) are time-varying variables and past contextual features may impact future flow. For example, sudden heavy rain may immediately reduce the crowd flow, but when the rain stops, the crowd flow may be larger than ever. Ke et al. \([14]\) use LSTM to capture temporal dependencies of features.

2) Feature Fusion Stage:

$E' \in \mathbb{R}^{T \times N \times P'}$ represents contextual representation after the context representation stage; $ST' \in \mathbb{R}^{T \times N \times Q'}$ denotes spatiotemporal representations learned by the neural network; the output of late fusing modeling is $O_l$. Context $E'$ can be fused with spatiotemporal representations $ST'$ by the following techniques (Fig. 1):

**Concatenate** is a widely used technique that combines different features. Chai et al. \([6]\) use a fully-connected layer as the embedding layer to represent features and then concatenate $E'$ with spatiotemporal latent representation $ST'$. The Concatenate formula is:

$$O_l = \text{Concat}(E'; ST') \quad (3)$$

**Add** does not expand the dimension of hidden states and thus has a lower computational cost. The formula of Add is listed below, $W_e$, $W_{st}$, and $b$ are learnable parameters, aligning the dimensions of $E'$ and $ST$.

$$O_l = E' \cdot W_e + ST' \cdot W_{st} + b \quad (4)$$

**Gating** \([11]\) considers features as the activation function of spatiotemporal features. It first transforms the feature to gating value $G$ and then uses $G$ to activate spatiotemporal features:

$$G = \sigma(E' \cdot W_e + b), O_l = \sigma(G \otimes ST') \quad (5)$$

where $W_e$ and $b$ are learnable parameters, “$\cdot$” and “$\otimes$” are the dot product and Hadamard product of two vectors. The
**TABLE II**

Late Fusion Details Consisting of Representation and Fusion Stage. Modeling techniques with ‘∗’ are newly found based on our analytic studies, which have never appeared in the literature to the best of our knowledge.

| Name                | Representation | Fusion | Literature |
|---------------------|----------------|--------|------------|
| Raw-Concat          | Raw            | Concat | [16]       |
| Raw-Add             | Raw            | Add    | [10], [26] |
| Raw-Gating          | Gating         |        | [11]       |
| Emb-Concat          | Embedding      | Concat | [6], [24]  |
| Emb-Add             | Embedding      | Add    | [25], [19] |
| Emb-Gating          | Gating         |        | [23]       |
| MultiEmb-Concat∗    | Concat         |        | [2], [5], [15], [23] |
| MultiEmb-Add∗       | Multiple Embedding | Add | -           |
| MultiEmb-Gating∗    | Gating         |        | -           |
| LSTM-Concat∗        | Concat         |        | -           |
| LSTM-Add            | LSTM           | Add    | [14]       |
| LSTM-Gating∗        | Gating         |        | -           |

**TABLE III**

Datasets Statistics

|            | Bike-sharing | Metro | EV         |
|------------|--------------|-------|------------|
| Chicago    | 2013-07-     | 2016-07- | 2018-03-  |
|           | 2014-09      | 2016-09 | 2018-05   |
| # Locations| 585          | 288    | 629        |
| # Weather States | 24      | 24    | 24         |
| # Holiday  | 3624         | 298    | 312        |
| # POIs Categories | 299     | 14    | 14         |

intuition of Gating is that features are like switches, and the crowd flows would be tremendously changed if a certain switch is activated.

In summary, by combining two stages, we list 12 (3*4) variants of Late Fusion Modelling techniques, as shown in Table II. It is worth noting that, with a comprehensive survey of the literature, we find that four variants never appeared in previous work. Meanwhile, we believe that these variants are reasonable context modeling techniques and we will also test them in our benchmark experiments.

IV. EMPIRICAL EXPERIMENT BENCHMARK

A. Datasets

We conduct experiments on three spatiotemporal crowd flow datasets including three scenarios (bike-sharing demand, metro passenger flow, and electric vehicle charging station usage). The dataset statistics are listed in Table III. Original records are processed at an interval of 30, 60, and 120 minutes.

1) Bike-sharing: The bike-sharing dataset is collected from Chicago open data portal. The dataset is open to everyone for non-commercial purposes. The time span of this dataset is more than one year and each piece of valid record contains the start station, start time, stop station, stop time, etc. We predict the number of bike-sharing demands in each station.

2) Metro: The metro dataset contains metro trip records in Shanghai, which are obtained by Shanghai Open Data Apps (SODA) challenge. The time span is three months. Each metro trip record has the check-in time, check-in station, check-out time, and check-out station. We predict the check-in flow amount for all the metro stations.

3) Electrical Vehicle (EV): The electrical vehicle (EV) dataset is collected from one major EV charging station operator in Beijing. The time span of the dataset is six months. This dataset contains the occupation situation at different time slots. Each record contains sensing time, available and occupied docks. We predict the number of docks in use for each station as it is the most important demand indicator of the charging stations.

4) Contextual Features: We collect weather data from the OpenWeatherMap website and its original interval is 60-minute. We regard the weather data measured by one meteorological station in each city as the weather data of the entire city. To get 120-minute weather data, we straightforwardly take the weather features in the first 60 minutes. To get 30-minute weather data, we approximate them by repeating the hourly weather data twice. For the holiday data, we parse holiday information by the chinese_calendar and workalendar packages. For the temporal position features, we transform the DayOfWeek and HourOfDay features by one-hot encoding. The POIs data in Shanghai and Beijing are collected with the developer APIs from the online map. The POIs data in Chicago are collected from the OpenStreetMap.

B. Implementation Details

To incorporate different temporal knowledge, the inputs of the STMGCN and STMeta networks consist of 17 historical observations, including six closeness records, seven daily records, and four weekly records. For the considerations of spatial knowledge, distance and correlation graphs are introduced. The distance graphs are calculated based on the euclidean distance. The correlation graphs are computed by the Pearson coefficient of the time series of stations.

The hidden states of STMeta and STMGCN backbone network both are 64 (the dimension of spatiotemporal representations). The degree of graph Laplacian is set to 1. We build three kinds of graphs as STMeta (i.e., proximity, function, and interaction graph). We get the output dimensions of embedding layers through search (Fig. 2). The output dimensions of single embedding layer (including Emb-Concat, Emb-Add, and Emb-Gating) are set to 16. The output dimensions of multiple embedding layer (including MultiEmb-Concat, MultiEmb-Add, and MultiEmb-Gating) are set to ‘8-1-8-8’, corresponding to weather, holiday, temporal position

---

3http://soda.data.sh.gov.cn/index.html
4https://openweathermap.org/history-bulk
5http://soda.data.sh.gov.cn/index.html
6https://openweathermap.org/history-bulk
7https://pypi.org/project/chinesecalendar/0.0.4
8https://github.com/peopledoc/workalendar
9https://map.baidu.com
10https://www.openstreetmap.org/
(a) Impact of different output dimensions of single embedding layer using Emb-Concat.

(b) Impact of different output dimensions of multiple embedding layers using MultiEmb-Concat. 8-1-8-8 denotes the embedding size applying to weather, holiday, temporal position, and POIs.

Fig. 2. Impact of different output dimensions of embedding layers based on STMGCN in the Metro datasets.

and POIs. Our experiment platform is a server with 8 CPU cores (11th Gen Intel(R) Core(TM) i7-11700K @ 3.60GHz), 32 GB RAM, and one GPU (NVIDIA TITAN Xp). We use python 3.6.5 with TensorFlow [36] on Ubuntu Linux release 5.11.1 (Core).

C. Model Variants

We implement STMGCN [7] and STMeta [8] as the spatiotemporal backbone networks in Fig. 1, since these two STCFP models have been verified to perform generally well in a recent large-scale benchmark study [8]. Early Joint Modeling and Late Fusion can be applied to both backbone networks. In addition, we implement an XGBoost-based STCFP model [32], by concatenating spatiotemporal and contextual features to further analyze the generalizability of contextual features on traditional machine learning models.

D. Evaluation Metrics

We exploit two widely used metrics, namely RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) to assess the performance of each method:

\[
\text{RMSE}(\hat{y}, \hat{y}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \quad (6)
\]

\[
\text{MAE}(\hat{y}, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} |(y_i - \hat{y}_i)| \quad (7)
\]

where \(y_i\) and \(\hat{y}_i\) are the ground truth and predict flows and \(N\) is the number of samples. Suppose that there is a set of approaches \(\mathcal{X}\) and several evaluation datasets \(\mathcal{D}\), avgNRMSE and avgNMAE are defined to assess the overall performance of each method. The avgNRMSE and avgNMAE of method \(x\) \((x \in \mathcal{X})\) are:

\[
\text{avgNRMSE}_x = \frac{\text{Average}_{d \in \mathcal{D}} \left( \frac{\text{RMSE}_{x,d}}{\min_{x' \in \mathcal{X}} \left( \text{RMSE}_{x',d} \right)} \right)}{\min_{x' \in \mathcal{X}} \left( \text{RMSE}_{x',d} \right)} \quad (8)
\]

\[
\text{avgNMAE}_x = \frac{\text{Average}_{d \in \mathcal{D}} \left( \frac{\text{MAE}_{x,d}}{\min_{x' \in \mathcal{X}} \left( \text{MAE}_{x',d} \right)} \right)}{\min_{x' \in \mathcal{X}} \left( \text{MAE}_{x',d} \right)} \quad (9)
\]

If \(\text{avgNRMSE}_x\) and \(\text{avgNMAE}_x\) are closer to 1, method \(x\) has superior performance across different datasets, indicating this method has better generalizability. The MAE results are consistent with the RMSE results. We show the RMSE results in the main manuscripts (Table IV, V, VI, VII, VIII) and the MAE results are listed in the appendix.

E. Result 1: Impact of Modeling Techniques

To analyze the generalizability of context modeling techniques, we compare 14 techniques elaborated in Section III based on STMeta [8] and STMGCN [7] backbone networks. The results are listed in Table [V] and Table [V] where the avgNRMSE is calculated to evaluate the generalizability. Emb-Gating and Raw-Gating have the lowest avgNRMSE based on STMGCN and STMeta backbone networks, respectively. We also display the training time of different context modeling techniques in Fig. 3 The training time of most techniques is
similar to \textit{No Context}, but \textit{EarlyConcat} brings more training parameters and thus is more time-consuming.

\subsection*{F. Result 2: Impact of Contextual Features}

To compare the generalizability of features, we design the ablation experiments by utilizing various combinations of features. We choose \textit{Raw-Gating} as the context modeling technique because it has good generalizability in Table IV and Table V. We conduct context combination experiments including weather, holiday, temporal position, and POIs data in metro passenger flow and electric vehicle charging demand prediction. We do not consider POIs in the bike flow prediction, as we have not obtained the POIs data for the bike-sharing dataset. The results are shown in Table VI, Table VII, and Table VIII. Each row is named as the features in consideration. For example, \textit{Wea-Holi} includes weather and holiday features.

\section*{V. Analysis on Benchmark Results}

\subsection*{A. Generalizability Analysis}

1) Late Fusion vs. Early Joint Modeling: In Table IV and Table V, late fusion methods are better than early joint modeling both in STMGCN and STMeta. Besides, when compared to \textit{No Context}, \textit{EarlyConcat} and \textit{EarlyAdd} are both worse. This means that it is better not to incorporate contextual features by early joint modeling. It is also worth noting that the late fusion by \textit{Gating} (e.g., \textit{Emb-Gating} and \textit{Raw-Gating}) performs the best for most settings, showing good generalizability across various application scenarios and different models.

2) Is context embedding always necessary? Context embedding is widely used in many STCFP studies [2], [5], [15], [25], [19]. According to Table IV and Table V, embedding layers may enhance model performance by mapping features into a low-dimension space for the methods when the fusion stage is \textit{Concat} or \textit{Add}. Comparing \textit{Raw-Concat} and \textit{Emb-Concat/MultiEmb-Concat}, the latter acquires lower avgNRMSE. Similar phenomena also occur in \textit{Raw-Add} vs. \textit{Emb-Add/MultiEmb-Add}. However, when considering the other fusion technique, \textit{Gating}, the observation is that embedding might not be useful. \textit{Raw-Gating}, which does not use embedding layers, still outperforms \textit{No Context} based on STMGCN and acquires the lowest avgRMSE based on STMeta. It is worth noting that, when applying embedding layers to the gated units, the performance becomes worse on STMeta. It shows that the embedding layers may work when the fusion methods are \textit{Concat} or \textit{Add}. But for \textit{Gating}, embedding layers may not be necessary and using \textit{Raw} features can already result in good and generalizable prediction performance.

3) Is past context beneficial? Weather features are time-varying, and thus temporal modeling units (e.g., LSTM) could be applied to model context temporal dependencies [14]. We leverage LSTM to capture temporal patterns from the weather context at the past six observations. From Table IV and Table V, we observe that, with STMGCN, using LSTM to model context features is even worse than \textit{No Context} which does not consider contextual features. LSTM is more suitable for STMeta but is still worse than \textit{Raw-Gating}. This inspires us that, for learning temporal context patterns, more advanced techniques than LSTM may be needed to investigate in the future.

4) Do more contextual features always result in better prediction? Intuitively, we may think the more features we use, the better prediction we get. However, our results show that this does not stand for most cases. Specifically, for different machine learning models, the highest average performance (avgRMSE) is all achieved when only holiday and temporal position (\textit{Holi-TP}) are considered. This is a surprising observation, as some widely-considered features like weather, would not bring generalizable improvements according to our results. The possible reasons may be: (1) the features are not fine-grained enough, e.g., all the locations in one city usually hold the same weather features; and (2) state-of-the-art context modeling techniques are still too simple to catch complicated patterns in features like weather. Hence, future studies on how to effectively incorporate contexts into spatiotemporal prediction models are still highly desired.

5) Do context modeling techniques increase the computational burden? If more features are considered, neural networks need more parameters to aggregate them, which may increase the computational burden. Fig. 3 displays the training time of different context modeling techniques based on STMeta. We observe that the training time of most techniques is similar to \textit{No Context}, except for \textit{EarlyConcat}. Intuitively, \textit{Concatenate} will increase feature dimensions, and corresponding increase the training time. But the training time of late concatenate methods (late fusion techniques whose fusion stage is \textit{Concatenate}) is smaller than \textit{EarlyConcat}. The reason is that the spatiotemporal networks in STMeta are more complex than the context representation methods of late fusion, dominating the training time of the whole model. To sum up, \textit{EarlyConcat} will increase the training time, and the remaining modeling techniques may not add significant computational burdens.

6) Do contextual features perform consistently in different interval prediction tasks? Weather features hardly improve the overall performance in these tasks. Their generalizable performance is consistently worse than \textit{No Context} in 30, 60, and 120-minute prediction tasks. It is worth noting that the temporal position features are with better generalizability in 30-minute prediction tasks than in 60 and 120-minute prediction tasks. It fits our expectation, as the temporal position becomes more fine-grained when the interval gets smaller, indicating more information. For example, the length of \textit{Hour/Day} features are 24 and 48 in 60 and 30-minute prediction tasks, respectively. Hence, we should pay more attention to temporal position features when conducting short-interval prediction tasks.

\subsection*{B. Guidelines and Insights}

In general, based on our benchmark, we find that adding features may not always increase STCFP prediction accuracy, and practitioners should carefully determine which features and modeling techniques are applied.
TABLE IV
30/60/120-MINUTE RMSE RESULTS OF DIFFERENT MODELING TECHNIQUES BASED ON STMGCN. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD. NO CONTEXT DOES NOT INCORPORATE ANY CONTEXTUAL FEATURES.

| STMGCN      | Bike  | Metro | EV    | avgNRMSE |
|-------------|-------|-------|-------|----------|
|             | 30    | 60    | 120   |          |
| No Context  | 2.282 | 2.767 | 4.334 | 84.89    |
| Early Joint Modeling |       |       |       |          |
| EarlyConcat | 2.240 | 2.849 | 4.309 | 85.14    |
| EarlyAdd    | 2.220 | 2.858 | 4.188 | 87.14    |
| Late Fusion  |       |       |       |          |
| Raw-Concat  | 2.224 | 2.79  | 4.245 | 80.09    |
| Raw-Add     | 2.409 | 3.091 | 4.350 | 88.20    |
| Raw-Gating  | 2.292 | 2.700 | 3.844 | 84.80    |
| Emb-Concat  | 2.246 | 2.951 | 4.414 | 83.91    |
| Emb-Add     | 2.160 | 2.773 | 4.193 | 87.67    |
| Emb-Gating  | 2.218 | 2.680 | 3.998 | 92.34    |
| MultiEmb-Concat | 2.181 | 2.798 | 4.317 | 83.91    |
| MultiEmb-Add | 2.208 | 2.909 | 4.173 | 87.67    |
| MultiEmb-Gating | 2.348 | 2.688 | 3.995 | 90.55    |
| LSTM-Concat | 2.192 | 2.833 | 4.478 | 94.57    |
| LSTM-Add    | 2.171 | 2.825 | 4.154 | 90.10    |
| LSTM-Gating | 2.227 | 2.701 | 3.959 | 85.14    |

TABLE V
30/60/120-MINUTE RMSE RESULTS OF DIFFERENT MODELING TECHNIQUES BASED ON STMETA. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD. NO CONTEXT DOES NOT INCORPORATE ANY CONTEXTUAL FEATURES.

| STMeta      | Bike  | Metro | EV    | avgNRMSE |
|-------------|-------|-------|-------|----------|
|             | 30    | 60    | 120   |          |
| No Context  | 2.211 | 2.740 | 3.830 | 77.62    |
| Early Joint Modeling |       |       |       |          |
| EarlyConcat | 3.173 | 3.027 | 4.731 | 119.5    |
| EarlyAdd    | 2.485 | 2.703 | 3.991 | 78.99    |
| Late Fusion  |       |       |       |          |
| Raw-Concat  | 2.229 | 2.665 | 3.834 | 83.51    |
| Raw-Add     | 2.205 | 2.632 | 3.797 | 84.63    |
| Raw-Gating  | 2.173 | 2.598 | 3.741 | 74.40    |
| Emb-Concat  | 2.199 | 2.630 | 3.840 | 77.35    |
| Emb-Add     | 2.124 | 2.701 | 3.794 | 80.04    |
| Emb-Gating  | 2.189 | 2.608 | 3.758 | 91.76    |
| MultiEmb-Concat | 2.133 | 2.593 | 3.800 | 78.91    |
| MultiEmb-Add | 2.254 | 2.634 | 3.776 | 88.05    |
| MultiEmb-Gating | 2.208 | 2.690 | 3.788 | 85.03    |
| LSTM-Concat | 2.116 | 2.594 | 3.777 | 77.23    |
| LSTM-Add    | 2.109 | 2.580 | 3.691 | 76.96    |
| LSTM-Gating | 2.167 | 2.585 | 3.648 | 78.99    |

1) Contextual features guidelines: (i) Weather, holiday, temporal position, and POIs could be considered in almost all kinds of STFCFP scenarios. However, using more contextual features may not always result in better predictions. (ii) The feature combination of holiday and temporal position can provide more generalizable beneficial information. Additionally, temporal position and holiday features are easy to access and thus are prior recommended. (iii) Temporal position features are especially critical for short-interval prediction tasks.

2) Modeling techniques guidelines: (i) Existing early joint modeling techniques (EarlyConcat and EarlyAdd) are not good enough to extract beneficial information from contextual features. That is to say, fusing context in the high-level layer of neural networks is a better choice rather than in the low-level layer. (ii) In the late fusion techniques, the embedding layers may work when the fusion methods are Concat and...
(iv) We recommend **Raw-Gating** and two state-of-the-art spatiotemporal neural network models. **Gating** is a highly recommended context fusion method because it has the best generalizability across three scenarios and two state-of-the-art spatiotemporal neural network models. It not only has a good and generalizable performance across different experiment scenarios but also does not need to tune many fusing parameters (e.g., embedding size).

### VI. Related Work

#### A. Time Series Prediction

Earlier research regarded the crowd flow prediction as a classic time series prediction problem. Hamed et al. [37] applied the Autoregressive Integrated Moving Average (ARIMA) to forecast short-term traffic on the highway. ARIMA is a linear model that assumes that future crowd flow is related to historical observation. But this assumption is not consistent with the actual crowd flow characteristics. Many nonlinear algorithms such as support vector machine [38], Markov random field [4], decision tree method [1] and bayesian network [39] model the temporal dependencies very well, yet...
it fails to capture spatial correlations and leverage contextual information.

B. Spatio-Temporal Prediction

With the enhancement of computing performance and the development of deep learning technology, Convolutional Neural Network (CNN) and Graph Convolutional Network (GCN) are used to model spatial dependency in crowd flow prediction problems. Zhang et al. [25] split the city traffic flow into grids according to time order, and then use CNN to capture the spatial dependency. Ke et al. [14] used convolution long-short term memory network (Conv-LSTM) to simultaneously capture the spatiotemporal dependencies of taxi demand. Geng et al. [7] use a multi-graph convolution model to capture varieties of spatial knowledge. Wang et al. [40] give a comprehensive review of recent progress for spatiotemporal prediction from the perspective of spatiotemporal data mining. Wang et al. [8] also provide an evaluation benchmark of the STCFP models and spatiotemporal knowledge. However, whereas these previous studies attempt to make full use of spatiotemporal correlations and build large-scale benchmarks, this paper focuses on exploring the generalizability of contextual features and their modeling techniques.

VII. DISCUSSION

A. Limitations

Due to the computation resource limitation, this work includes three typical datasets (i.e., bike-sharing demand, metro flow, and electric vehicle usage) and three machine learning methods (one for traditional machine learning and two for deep learning). It is expected to run more experiments based on more datasets and models to further consolidate our observations and findings. It is worth noting that, for one dataset, one STCFP deep learning method, and one time interval, our benchmark needs to train 31 models under various configurations (16 for different contextual feature combinations, and 15 for different context modeling techniques). This leads to the training of several hundred models in total. Because of such a large number of configurations required for the benchmark, we carefully select the effective deep learning methods that have been validated to hold good generalizability across diverse STCFP applications [8].

B. Future Work

1) Advanced modeling techniques: Raw-Gating, with good generalizability, is a remarkable technique to model contextual features. We suggest leveraging Raw-Gating as the default modeling technique. However, it may still worsen the performance compared to No Context in some (although rarely) specific tasks (e.g., 60-minute metro flow prediction based on STMGCN). In other words, we have not found a single context modeling technique that can improve the STCFP performance in every one of our experimental scenarios. Hence, an advanced context modeling technique is still urgently demanded to benefit the STCFP research community.

2) More datasets and models beyond crowd flow prediction: In addition to urban crowd flow, many other applications are highly dependent on contextual factors. Hence, we plan to extend our benchmark scenarios to a broader scope of spatiotemporal prediction scenarios such as traffic speed and congestion.

VIII. CONCLUSION

In this paper, we focus on exploring the generalizability of both contextual features and context modeling techniques in the STCFP problem. We conduct both analytical and experimental studies. In the analytical studies, we investigate various contextual features and modeling techniques in the literature and develop a comprehensive taxonomy of context
modeling techniques. In the experimental benchmark, we analyze the generalizability of different contextual features including weather, holiday, temporal position, and POIs. We also analyze the generalizability of diverse modeling techniques based on state-of-the-art spatiotemporal prediction models, i.e., XGBoost [32], STMGCN [27], and STMeta [8]. Besides, we develop an experimental platform composed of large-scale spatiotemporal crowd flow data, contextual data, and state-of-the-art spatiotemporal prediction models. We also offer several suggestions about incorporating contextual factors for practitioners who want to build STCFP applications.

REFERENCES

[1] Y. Li, Y. Zheng, H. Zhang, and L. Chen, “Traffic prediction in a bike-sharing system,” in Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS ’15, 2015, pp. 1–10.
[2] Y. Liu, Z. Liu, and R. Jia, “DeepPF: A deep learning based architecture for metro passenger flow prediction,” Transportation Research Part C: Emerging Technologies, vol. 101, pp. 18–34, 2019.
[3] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” in International symposium on handheld and ubiquitous computing, 1999, pp. 304–307.
[4] M. X. Huang, Y. Zheng, and A. K. Singh, “Fcf: Forecasting citywide crowd flows based on big data,” in Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016.
[5] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, “GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction,” in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018, pp. 3428–3434.
[6] D. Chai, L. Wang, and Q. Yang, “Bike flow prediction with multi-graph convolutional networks,” in Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2018, p. 397–400.
[7] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 3656–3663, 2019.
[8] B. Wang, D. Chai, X. Liu, L. Chen, and K. Chen, “Exploring the generalizability of spatio-temporal traffic prediction: Meta-modeling and an analytic framework,” IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 2021.
[9] Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye, and W. Lv, “The Simpler The Better: A Unified Approach to Predicting Original Taxi Demands based on Large-Scale Online Platforms,” in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’17, 2017, pp. 1653–1662.
[10] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, “Dnn-based prediction model for spatio-temporal data,” in Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016, pp. 1–4.
[11] J. Zhang, Y. Zheng, J. Sun, and D. Qi, “Flow Prediction in Spatio-Temporal Networks Based on Multitask Deep Learning,” IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 2019.
[12] Z. Yang, J. Hu, Y. Shu, P. Cheng, J. Chen, and T. Moscibroda, “Mobility Modeling and Prediction in Bike-Sharing Systems,” in Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys ’16, 2016, pp. 165–178.
[13] Y. Li and Y. Zheng, “Citywide bike usage prediction in a bike-sharing system,” IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 6, pp. 1079–1091, 2020.
[14] J. Ke, H. Zheng, H. Yang, and X. M. Chen, “Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach,” Transportation Research Part C: Emerging Technologies, vol. 85, pp. 591–608, 2018.
[15] D. Wong, W. Cao, L. Li, and J. Ye, “DeepSD: Supply-Demand Prediction for Online Car-Hailing Services Using Deep Neural Networks,” in 2017 IEEE 33rd International Conference on Data Engineering (ICDE), 2017, pp. 243–254.
[16] L. Zhu and N. Laptev, “Deep and confident prediction for time series at uber,” in 2017 IEEE International Conference on Data Mining Workshops (ICDMW), 2017, pp. 103–110.
[17] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, “Deep multi-view spatial-temporal network for taxi demand prediction,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.
[18] A. Saadallah, L. Moreira-Matias, R. Sousa, J. Khairi, E. Jenelius, and J. Gama, “Bright—drift-aware demand predictions for taxi networks,” IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 2, pp. 234–245, 2020.
[19] X. Yi, Z. Duan, T. Li, T. Li, J. Zhang, and Y. Zheng, “Citytraffic: Modeling citywide traffic via neural memorization and generalization approach,” in Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2665–2671.
[20] C. Zheng, X. Fan, C. Wang, and J. Qi, “Gman: A graph multi-attention network for traffic prediction,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, pp. 1234–1241, 2020.
[21] R. Barnes, S. Buthpiyta, J. Cook, A. Fabrikant, A. Tomkins, and F. Xu, “Bustr: Predicting bus travel times from real-time traffic,” in KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020. ACM, 2020, pp. 3243–3251.
[22] C. Zheng, X. Fan, C. Wen, L. Chen, C. Wang, and J. Li, “Deepst: Mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 3744–3755, 2020.
[23] X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, and Y. Zheng, “Traffic flow forecasting with spatial-temporal graph diffusion network,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 17, pp. 15008–15015, 2021.
[24] H. Yuan, G. Li, Z. Bao, and L. Feng, “An effective joint prediction model for travel demands and traffic flows,” in 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021, pp. 348–359.
[25] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in Proceedings of the Thirty-First International Conference on Scientific and Statistical Database Management, 2019, p. 1655–1661.
[26] A. Zonoozi, J. jae Kim, X.-L. Li, and G. Cong, “Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns,” in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, 7 2018, pp. 3732–3738.
[27] K. Chen, F. Chen, B. Lai, Z. Jin, Y. Liu, K. Li, L. Wei, P. Wang, Y. Tang, J. Huang, and X. S. Hua, “Dynamic spatio-temporal graph-based cns for traffic flow prediction,” IEEE Access, vol. 8, pp. 185136–185145, 2020.
[28] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng, “Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks,” IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 07 2020.
[29] R. Jiang, Z. Cai, Z. Wang, C. Yang, Z. Fan, Q. Chen, K. Tsubouchi, X. Song, and R. Shibasaki, “Deepcrowd: A deep model for large-scale citywide crowd density and flow prediction,” IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 2021.
[30] F. Wu, H. Wang, and Z. Li, “Interpreting traffic dynamics using ubiquitous urban data,” in Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016.
[31] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, “Discovering urban functional zones using latent activity trajectories,” IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 3, pp. 712–725, 2014.
[32] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’16. New York, NY, USA: ACM, 2016, p. 785–794.
[33] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, “Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 1020–1027, 2019.
[34] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, and Z. Li, “Modeling spatial-temporal dynamics for traffic prediction,” ArXiv, vol. abs/1803.01254, 2018.
[35] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga,
S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, ser. OSDI’16. USA: USENIX Association, 2016, p. 265–283.

[37] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, “Short-term prediction of traffic volume in urban arterials,” Journal of Transportation Engineering, vol. 121, no. 3, pp. 249–254, 1995.

[38] Y. Cong, J. Wang, and X. Li, “Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm,” Procedia Engineering, vol. 137, no. 1, pp. 59–68, 2016.

[39] J. Wang, W. Deng, and Y. Guo, “New bayesian combination method for short-term traffic flow forecasting,” Transportation Research Part C: Emerging Technologies, vol. 43, pp. 79–94, 2014.

[40] S. Wang, J. Cao, and P. Yu, “Deep learning for spatio-temporal data mining: A survey,” IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 2020.
APPENDIX

We here list the MAE results of different context modeling techniques (Table IX, X) and the MAE results of different combination of contextual features (Table XI, XII, XIII).
### TABLE IX
30/60/120-minute MAE results of different modeling techniques based on STMGCN. The best results are highlighted in bold. No Context does not incorporate any contextual features.

| Context            | Bike       | Metro     | EV         | avgNMAE   |
|--------------------|------------|-----------|------------|-----------|
| No Context         | 1.406      | 1.575     | 2.192      | 1.030     |
| Early Joint Modeling |           |           |            |           |
| EarlyConcat        | 1.397      | 1.576     | 2.216      | 0.328     |
| EarlyAdd           | 1.395      | 1.582     | 2.239      | 0.327     |
| Late Fusion        |            |           |            |           |
| Raw-Concat         | 1.348      | 1.569     | 2.216      | 0.381     |
| Raw-Add            | 1.390      | 1.636     | 2.379      | 0.409     |
| Raw-Gating         | 1.400      | 1.556     | 2.211      | 0.338     |
| Emb-Concat         | 1.376      | 1.612     | 2.457      | 0.351     |
| Emb-Add            | 1.366      | 1.576     | 2.169      | 0.338     |
| Emb-Gating         | 1.408      | 1.559     | 2.111      | 0.345     |
| MultiEmb-Concat    | 1.375      | 1.571     | 2.219      | 0.351     |
| MultiEmb-Add       | 1.431      | 1.609     | 2.171      | 0.340     |
| MultiEmb-Gating    | 1.397      | 1.559     | 2.196      | 0.330     |
| LSTM-Concat        | 1.426      | 1.586     | 2.217      | 0.335     |
| LSTM-Add           | 1.366      | 1.576     | 2.187      | 0.321     |
| LSTM-Gating        | 1.439      | 1.562     | 2.168      | 0.343     |

### TABLE X
30/60/120-minute MAE results of different modeling techniques based on STMeta. The best results are highlighted in bold. No Context does not incorporate any contextual features.

| Context            | Bike       | Metro     | EV         | avgNMAE   |
|--------------------|------------|-----------|------------|-----------|
| No Context         | 1.374      | 1.521     | 2.093      | 1.048     |
| Early Joint Modeling |           |           |            |           |
| EarlyConcat        | 1.961      | 1.791     | 2.521      | 0.683     |
| EarlyAdd           | 1.644      | 1.518     | 2.147      | 0.380     |
| Late Fusion        |            |           |            |           |
| Raw-Concat         | 1.382      | 1.483     | 2.081      | 0.340     |
| Raw-Add            | 1.417      | 1.491     | 2.053      | 0.366     |
| Raw-Gating         | 1.394      | 1.457     | 2.061      | 0.338     |
| Emb-Concat         | 1.372      | 1.490     | 2.063      | 0.345     |
| Emb-Add            | 1.295      | 1.515     | 2.053      | 0.341     |
| Emb-Gating         | 1.349      | 1.466     | 2.066      | 0.347     |
| MultiEmb-Concat    | 1.350      | 1.451     | 2.062      | 0.341     |
| MultiEmb-Add       | 1.458      | 1.494     | 2.060      | 0.342     |
| MultiEmb-Gating    | 1.360      | 1.502     | 2.063      | 0.342     |
| LSTM-Concat        | 1.380      | 1.456     | 2.070      | 0.340     |
| LSTM-Add           | 1.334      | 1.440     | 2.064      | 0.332     |
| LSTM-Gating        | 1.390      | 1.450     | 2.012      | 0.333     |
### TABLE XI

30-minute MAE results of different contextual features. The best results are highlighted in bold. (W: Weather; H: Holiday; TP: Temporal Position; POI: Point of Interest)

| Model | Bike | Metro | EV | avgNMAE |
|-------|------|-------|----|---------|
| XGBoost | Bike | Metro | EV | avgNMAE |
| No Context | 1.532 | 39.41 | 0.336 | 1.016 |
| W: Weather | 0.655 | 1.005 |
| H: Holiday | 1.683 | 73.95 | 0.456 | 1.026 |
| TP: Temporal Position | 1.683 | 73.95 | 0.456 | 1.026 |
| POI: Point of Interest | 1.683 | 73.95 | 0.456 | 1.026 |

### TABLE XII

60-minute MAE results of different contextual features. The best results are highlighted in bold. (W: Weather; H: Holiday; TP: Temporal Position; POI: Point of Interest)

| Model | Bike | Metro | EV | avgNMAE |
|-------|------|-------|----|---------|
| XGBoost | Bike | Metro | EV | avgNMAE |
| No Context | 1.737 | 71.33 | 0.458 | 1.026 |
| W: Weather | 0.655 | 1.005 |
| H: Holiday | 1.683 | 73.95 | 0.456 | 1.026 |
| TP: Temporal Position | 1.683 | 73.95 | 0.456 | 1.026 |
| POI: Point of Interest | 1.683 | 73.95 | 0.456 | 1.026 |

### TABLE XIII

120-minute MAE results of different contextual features. The best results are highlighted in bold. (W: Weather; H: Holiday; TP: Temporal Position; POI: Point of Interest)

| Model | Bike | Metro | EV | avgNMAE |
|-------|------|-------|----|---------|
| XGBoost | Bike | Metro | EV | avgNMAE |
| No Context | 2.246 | 137.6 | 0.695 | 1.032 |
| W: Weather | 0.655 | 1.005 |
| H: Holiday | 2.178 | 139.8 | 0.694 | 1.027 |
| TP: Temporal Position | 2.178 | 139.8 | 0.694 | 1.027 |
| POI: Point of Interest | 2.178 | 139.8 | 0.694 | 1.027 |