# Self-Supervised Deconfounding Against Spatio-Temporal Shifts: Theory and Modeling

Jiahao Ji, Wentao Zhang, Jingyuan Wang SCSE, Beihang University Beijing, China {jiahaoji, zhangwt97, jywang}@buaa.edu.cn Yue He
CST, Tsinghua University
Beijing, China
heyuethu@mail.tsinghua.edu.cn

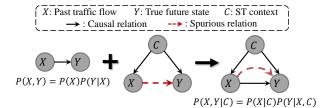
Chao Huang
CS & IDS, University of Hong Kong
Hong Kong, China
chaohuang75@gmail.com

Abstract—As an important application of spatio-temporal (ST) data, ST traffic forecasting plays a crucial role in improving urban travel efficiency and promoting sustainable development. In practice, the dynamics of traffic data frequently undergo distributional shifts attributed to external factors such as time evolution and spatial differences. This entails forecasting models to handle the out-of-distribution (OOD) issue where test data is distributed differently from training data. In this work, we first formalize the problem by constructing a causal graph of past traffic data, future traffic data, and external ST contexts. We reveal that the failure of prior arts in OOD traffic data is due to ST contexts acting as a confounder, i.e., the common cause for past data and future ones. Then, we propose a theoretical solution named Disentangled Contextual Adjustment (DCA) from a causal lens. It differentiates invariant causal correlations against variant spurious ones and deconfounds the effect of ST contexts. On top of that, we devise a Spatio-Temporal sElfsuperVised dEconfounding (STEVE) framework. It first encodes traffic data into two disentangled representations for associating invariant and variant ST contexts. Then, we use representative ST contexts from three conceptually different perspectives (i.e., temporal, spatial, and semantic) as self-supervised signals to inject context information into both representations. In this way, we improve the generalization ability of the learned context-oriented representations to OOD ST traffic forecasting. Comprehensive experiments on four large-scale benchmark datasets demonstrate that our STEVE consistently outperforms the state-of-the-art baselines across various ST OOD scenarios.

Index Terms—Spatio-temporal forecasting, Urban computing

# I. INTRODUCTION

With the ubiquitous use of GPS-enabled mobile devices and sensors, a huge volume of spatio-temporal (ST) data is emerging from a variety of domains, *e.g.*, urban transportation. These ST data can support a growing number of applications. One important application is the Intelligent Transportation Systems (ITS), which has gained substantial attention in both academia and industry [1]. A pivotal facet of ITS revolves around ST traffic forecasting, aimed at accurate prediction of future traffic conditions (*e.g.*, traffic flow, traffic demand). Its significance reverberates across various urban applications, such as enhancing traffic efficiency through congestion management [2], [3], and promoting environmentally friendly commuting through bikesharing initiatives [4], [5]. Due to the impact of space- and time-dependent external factors, such as time and weather variations, ST traffic forecasting often faces the out-of-distribution (OOD)



(a) The proposed causal graph



(b) Intuitive example of OOD traffic forecasting

Fig. 1. Illustration of the OOD traffic forecasting problem and the causality behind it. (a) The causal graph among input X, output Y, and confounder C. The correlation between X and Y contains both causal and spurious relations. (b) Evening peak hours (as a confounder) on workdays produce spurious correlations between two distant road segments (road 1 and road 3), which disappear on holidays. Meanwhile, the causal relation between adjacent road segments (road 1 and road 2) is stable on both workdays and holidays.

problem. That is, the distribution of urban traffic ST data undergoes a change from the training phase to the test phase.

For example, the distribution shift can emerge when examining traffic patterns on holidays versus routine workdays. To investigate the OOD problem in urban traffic data, we establish a causal graph inspired by [4] to formalize the causal structure between historical traffic flows X, the true future state Y, and external ST contexts C. As shown in Fig. 1(a), C affects both X and Y in the causal graph, called the confounder of X and Y. Due to the changed dependence between C and Xin heterogeneous environments, distinct ST contexts would introduce variant spurious correlations into traffic data. Taking Fig. 1(b) as an example, evening peak hours (a kind of ST context) on workdays produce spurious correlations between two distant road segments (road 1 and road 3), which disappear on holidays. However, the causal relation between adjacent road segments (road 1 and road 2) is stable on both workdays and holidays. A trustworthy traffic forecasting model should maintain its effective prediction in different environments, that is to achieve the OOD generalization.

The OOD generalization in ST traffic forecasting is confronted with two main challenges. **First**, existing methods [1], [6]–[8] overlook the negative effects of spurious correlations in training data. They assume the training and test data are independent and identically distributed (i.i.d.), leading to an inability to differentiate between variant spurious correlations and invariant causal relations. Consequently, these methods fail to eliminate spurious correlations and show unstable performance in OOD test data. **Second**, ST contexts play a significant role the traffic data generation. However, available ST contexts are usually limited due to constraints in ST data collection and the uncertainty of traffic scenarios. Therefore, it is vital to leverage the limited context information to attain the model's generalization ability for unseen ST contexts.

To address the challenges of ST forecasting for OOD urban traffic data, we propose a Spatio-Temporal sElf-superVised dEconfounding (STEVE) framework that incorporates causal inference theory into ST dependency modeling. First, we put forward a novel disentangled contextual adjustment (DCA) in ST traffic scenarios following the backdoor adjustment theory [9]. It eliminates the spurious correlations by decoupling traffic flows and ST contexts in the causal graph through a do intervention. Then, to inject context information into our model, we design three self-supervised auxiliary tasks by virtue of spatial location, temporal index, and traffic capacity. This helps our STEVE jointly characterize the latent ST contexts and capture the causal relations that affect traffic generation under the principle of DCA. To sum up our contributions,

- We are pioneering to investigate the problem of spatiotemporal forecasting for OOD urban traffic data caused by ST contexts, and provide a casual interpretation to formalize this widespread problem in development.
- We propose a novel causal-informed spatio-temporal framework STEVE that removes the variant spurious correlations brought by ST contexts through a do-intervention based disentangled contextual adjustment.
- We subtly design three self-supervised tasks that characterize
  the latent ST contexts from the partial context information
  in pure observational traffic data, to align traffic representations with corresponding ST contexts. We believe this
  self-supervised deconfounding paradigm can offer insights
  into other areas involving latent confounders.
- Extensive experiments on four real-world large-scale traffic datasets show the superiority of our STEVE across various OOD scenarios of traffic forecasting. Furthermore, a case study confirms that our model can learn the latent distribution of unseen contexts by using partial ST contexts.

### II. PROBLEM FORMULATION

**Basic Concepts.** This paper aims to address the network-based ST traffic forecasting problem. The network is defined as  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ . It consists of nodes (e.g., spatial regions, road segments) denoted by  $\mathcal{V} = \{v_n | 1 \leq n \leq N\}$ , edges denoted by  $\mathcal{E}$ , and a binary adjacent matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  showing which nodes are connected. For example, We denote

 $\label{thm:list_of_major} TABLE\ I$  List of major notations and their definitions.

Notations	Definitions
$\overline{X}$	Random variable of past traffic data (input)
Y	Random variable of future traffic data (output)
C	Random variable of ST context
$C_I, C_V$	Random variable of invariant/variant ST context
$\mathcal C$	All possible ST context
$\mathcal{C}_I,\mathcal{C}_V$	All possible invariant/variant ST context
$do(\cdot)$	do-operator in causal inference
$\mathcal{G}$	The traffic network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ with node set $\mathcal{V}$
	and edge set $\mathcal{E}$
$A_{N}$	The adjacency matrix of traffic network $\mathcal{G}$
N	Number of nodes, i.e., $ \mathcal{V}  = N$
T	Window size of past data to be considered
d	Number of feature channels of traffic data
D	Dimension of traffic representation
$oldsymbol{X}_t$	Traffic data at the $t$ -th time step
$\boldsymbol{Z}_t$	Traffic representation at the $t$ -th time step
$\mathcal{X}$	Traffic data of recent $T$ past time step
${\mathcal Z}$	Traffic representation over multiple time steps
$\mathcal{Z}_I,\mathcal{Z}_V$	Traffic representation w.r.t. invariant/variant ST context
$\alpha_1, \alpha_2$	Prior probability of invariant/variant ST context

the traffic flow data observed on  $\mathcal{G}$  at current time step t as  $\boldsymbol{X}_t \in \mathbb{R}^{N \times d}$ , where d indicates the number of feature channels. Then,  $\mathcal{X} = (\boldsymbol{X}_{t-T+1}, \dots, \boldsymbol{X}_t) \in \mathbb{R}^{T \times N \times d}$  denotes the past T observations, and  $\boldsymbol{Y} = \boldsymbol{X}_{t+1}$  denotes the future traffic state. The major notations in this paper are listed in Tab. I.

**Traditional Traffic Forecasting** models assume the traffic data generation process is P(X,Y) = P(X)P(Y|X), as in the first graphical model in Fig. 1(a). The relevant forecasting problem is formulated as: given data drawn from training distribution  $P_{tr}(X,Y)$ , find an optimal model  $f_{\Theta^*}$  which can generalize best on data drawn from test distribution  $P_{te}(X,Y)$  w.r.t. loss function  $\ell$ :

$$f_{\Theta^*} = \arg\min_{f_{\Theta}} \mathbb{E}_{(\mathcal{X}, \mathbf{Y}) \sim P_{te}(X, Y)} [\ell(f_{\Theta}(\mathcal{X}), \mathbf{Y})]$$
  
s.t.  $P_{tr}(X, Y) = P_{te}(X, Y).$  (1)

However, as in the third graphical model in Fig. 1(a), the real traffic data generation process should be P(X,Y|C) = P(X|C)P(Y|X,C). The *traditional* problem formulation overlooks ST contexts C acting as a confounder. This can lead to unsatisfactory test performance once the distribution of C is changed. Thus, a **re-formulation** of this problem is necessary.

**OOD Traffic Forecasting.** Given data drawn from training distribution  $P(X,Y|C=\mathcal{C}_{tr})$  that is affected by training ST context  $\mathcal{C}_{tr}$ , we aim to find an optimal model  $f_{\Theta^*}$  which can generalize best on data of test distribution  $P(X,Y|C=\mathcal{C}_{te})$  that is different from  $P(X,Y|C=\mathcal{C}_{tr})$ :

$$f_{\Theta^*} = \arg\min_{f_{\Theta}} \mathbb{E}_{(\mathcal{X}, \mathbf{Y}) \sim P(X, Y | C = \mathcal{C}_{te})} [\ell(f_{\Theta}(\mathcal{X}), \mathbf{Y})]$$
  
s. t. 
$$P(X, Y | C = \mathcal{C}_{tr}) \neq P(X, Y | C = \mathcal{C}_{te}),$$
 (2)

where  $\ell$  is a loss function that measures the error between the predicted traffic state and ground truth.

Note Eq. (2) differs from Eq. (1) in two aspects: i) modeling of the impact of C on traffic data generation, ii) OOD assump-

tion described by  $P(X,Y|C=\mathcal{C}_{tr}) \neq P(X,Y|C=\mathcal{C}_{te})$ . Thus, previous models [1], [7], [10] trained via Eq. (1) cannot generalize well to the OOD Traffic Forecasting task.

# III. DISENTANGLED CONTEXTUAL ADJUSTMENT AND MODEL INSTANTIATION

This section first provides a theoretical scheme, called Disentangled Contextual Adjustment (DCA) for OOD traffic forecasting via causal intervention. Then, we instantiate the DCA as a learning model called STEVE in a principled way.

# A. Theoretical Scheme

One possible approach to solving the OOD problem is to learn causal relations that are stable across different data distributions [11]. To obtain a model based on causal relations and remove the spurious correlation brought by confounder C, we propose to intervene X by applying do-operator to variable X. The do-operator acts as intervention [12]. For example, do(X=1) means to actively set the value of X to 1, regardless of its passively observed value. In this way, the do-operator erases all arrows that come into X, i.e.,  $C \to X$  in Fig. 2. Once the link  $C \to X$  is cut off, the spurious correlation between X and Y disappears. Therefore, we obtain an OOD traffic forecasting model approximating  $P_{\Theta}(Y|do(X))$ , where  $\Theta$  is the model parameters.

The standard approach to intervening X is conducting a randomized controlled trial by collecting traffic data of any possible ST context, in which case  $P_{\Theta}(Y|do(X))$  equals  $P_{\Theta}(Y|X)$ . Such intervention is impossible because we cannot control the ST context. Fortunately, the back-door adjustment [9] provides a statistical estimation of  $P_{\Theta}(Y|do(X))$  using observed data. It first stratifies the confounder variable into discrete types. Then, it computes a weighted average of those types, where each type is weighted according to its proportion or prior probability. Specifically, we stratify the ST context into K discrete types, i.e.,  $\mathcal{C} = \{\mathcal{C}_k | 1 \leq k \leq K\}$ . Then, through the basic rules induced by the do-operator, we can estimate  $P_{\Theta}(Y|do(X))$  by:

$$P_{\Theta}(Y|do(X)) = \sum_{k=1}^{K} P_{\Theta}(Y|X, C = \mathcal{C}_k) P(C = \mathcal{C}_k).$$
 (3)

However, achieving the above backdoor adjustment is intractable because ST contexts are unobserved and their number K can be very large in the real world.

To address this challenge, we propose a disentangled contextual adjustment (DCA). The main idea is to disentangle ST contexts into two independent types: invariant and variant. The invariant contexts are responsible for ST contexts that do not change with the environment. The variant contexts are responsible for ST contexts that change fast across space and time like traffic jams and weather.

**Theorem 1.** The disentangled contextual adjustment (DCA) can estimate  $P_{\Theta}(Y|do(X))$  via

$$P_{\Theta}(Y|do(X)) = P(C = \mathcal{C}_I)P_{\Theta}(Y|X, C = \mathcal{C}_I) + P(C = \mathcal{C}_V)P_{\Theta}(Y|X, C = \mathcal{C}_V),$$
(4)

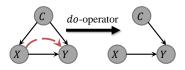


Fig. 2. Causal graph of our model  $P_{\Theta}(Y|do(X))$  that removes spurious correlations caused by C (red dashed arrow) via do-operator on X.

where the ST context is stratified into two **independent** types  $\mathcal{C}_I$  and  $\mathcal{C}_V$  constrained by  $\mathcal{C}_I \cup \mathcal{C}_V = \mathcal{C}$  and  $\mathcal{C}_I \cap \mathcal{C}_V = \emptyset$ . Specifically,  $\mathcal{C}_I = \{\mathcal{C}_{I_k} | I_1 \leq I_k \leq I_K\}$  denotes invariant contexts.  $\mathcal{C}_V = \{\mathcal{C}_{V_k} | V_1 \leq V_k \leq V_K\}$  denotes variant contexts. For the total type number K of ST contexts, we have  $K = I_K + V_K$ .

Note that the focus of disentanglement in DCA is on variables referred to as ST contexts. Each ST context is treated as an individual variable rather than a single value, such as weather. The terms "invariant" and "variant" are used to describe these context variables. These two types of context variables can cover all ST contexts in Eq. (3) based on the assumption below.

**Assumption 1** (ST context category). For K types of ST contexts in Eq. (3), each type  $C_k$  may involve both invariant context  $C_{I_k}$  and variant context  $C_{V_k}$  according to membership degree  $d_{I_k}$  and  $d_{V_k}$  constrained by  $d_{I_k} + d_{V_k} = 1$ . We treat  $C_k$  as invariant if  $d_{I_k} \geq d_{V_k}$ , otherwise it is variant.

We use  $C_I$  and  $C_V$  to denote the collective random variables of  $C_I$  and  $C_V$ . According to the laws of probability theory, we can have the following two rules:

Rule 1. Variable expansion of ST context:

$$P(C_I) = P(C = \mathcal{C}_I), P(C_V) = P(C = \mathcal{C}_V). \tag{5}$$

Rule 2. Conditional probability of ST context:

$$P(C_I = C_{I_k}) = \frac{P(C = C_{I_k})}{P(C_I)}, P(C_V = C_{V_k}) = \frac{P(C = C_{V_k})}{P(C_V)}.$$
(6)

With the above two rules, we can prove Theorem 1. Please refer to Sec. VI for details.

**Remark.** Recall our proposed DCA in Eq. (4). We can rewrite the data distribution on the right-hand side as a traffic data forecasting model:

$$P_{\Theta}(Y|do(X)) = \alpha_1 \cdot f_{\theta_1}(X, \mathcal{C}_I) + \alpha_2 \cdot f_{\theta_2}(X, \mathcal{C}_V), \quad (7)$$

where  $f_{\theta_1}(\cdot)$  parameterizes the invariant contextual conditional probability  $P_{\Theta}(Y|X,C=\mathcal{C}_I)$ , and  $f_{\theta_2}(\cdot)$  parameterizes the variant contextual conditional probability  $P_{\Theta}(Y|X,C=\mathcal{C}_V)$ .  $\alpha_1,\alpha_2$  denote the prior probabilities  $P(C=\mathcal{C}_I),P(C=\mathcal{C}_V)$ , and  $\Theta=\{\theta_1,\theta_2\}$ . As long as we implement the functions  $f_{\theta_1}(\cdot)$  and  $f_{\theta_2}(\cdot)$ , we can make the causal effect  $X\to Y$  free from the confounding effect of C. In this way, our approach can learn robust causal relations to achieve OOD generalization.

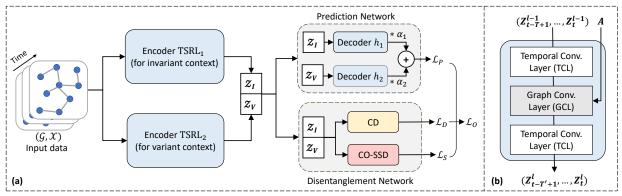


Fig. 3. (a) Illustration of our STEVE. The input traffic data are fed into two Traffic Sequence Representation Learning (TSRL) encoders to produce traffic representation  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$ . Then, they are decoupled by Contextual Disentanglement (CD) and aligned with invariant and variant ST contexts through a Context-Oriented Self-Supervised Deconfounding (CO-SSD) module. Note that the Gradient Reversal Layer (GRL) is embedded in CO-SSD. Representations  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  are used for traffic forecasting by a prediction network, where  $\alpha_1$  and  $\alpha_2$  are learnable priors for invariant and variant ST contexts. (b) Illustration of the building block of TSRL, *i.e.*, the "sandwich" structure. Specifically, we use two blocks to construct TSRL. l is the layer number.

### B. Model Design

To implement Eq. (7), we propose a ST self-supervised deconfounding model STEVE designed as follows:

$$\hat{\mathbf{Y}} = \alpha_1 \cdot h_1(\mathcal{Z}_I) + \alpha_2 \cdot h_2(\mathcal{Z}_V), \tag{8}$$

where  $\hat{Y}$  is the prediction of the future traffic state. The learnable vector  $\boldsymbol{\alpha}=(\alpha_1,\alpha_2)^{\top}$  parameterizes the prior probabilities  $P(C=\mathcal{C}_I), P(C=\mathcal{C}_V)$  with  $\alpha_1+\alpha_2=1$ . We implement the vector as  $\boldsymbol{\alpha}=\operatorname{SoftMax}(u((\mathcal{Z}_I,\mathcal{Z}_V)^{\top}))$ , where  $u(\cdot)$  is a linear transformation.  $h_1(\mathcal{Z}_I)$  and  $h_2(\mathcal{Z}_V)$  are the implementation of functions  $f_{\theta_1}(X,\mathcal{C}_I)$  and  $f_{\theta_2}(X,\mathcal{C}_V)$  in Eq. (7).  $h(\cdot)$  is implemented by a 1-D convolution network followed by an MLP.  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  are representations of traffic sequence. They contain information of invariant and variant ST contexts, respectively. The overall framework of our STEVE is depicted in Fig. 3(a).

Next, we employ a traffic sequence representation learning module to encode input data  $\mathcal{X}$  into  $\mathcal{Z}$ , and utilize a contextual disentanglement module to decouple  $\mathcal{Z}$  into  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$ .

1) Traffic Sequence Representation Learning: The TSRL module aims to transform the input traffic sequence  $\mathcal X$  into a representation  $\mathcal Z$ . A temporal convolutional layer and a graph convolution layer are employed by the TSRL to exploit temporal and spatial dependencies.

**Temporal Convolutional Layer (TCL).** We take traffic flow sequence  $\mathcal{X} = (X_{t-T+1}, \dots, X_t) \in \mathbb{R}^{T \times N \times d}$  as the input data of TCL. We employ a 1-D causal convolution along the time dimension [7] to implement TCL. Our TCL then outputs a time-aware traffic representation:

$$(\boldsymbol{H}_{t-T_1+1}, \dots, \boldsymbol{H}_t) = \text{TCL}(\boldsymbol{X}_{t-T+1}, \dots, \boldsymbol{X}_t), \quad (9)$$

where  $H_t \in \mathbb{R}^{N \times D}$  is the traffic representation matrix at time step t, and  $T_1$  is the length of the output sequence. Here, N is the node number of our input network, and D is the representation dimension.

**Graph Convolutional Layer (GCL).** We take the output of TCL as input. Our GCL is implemented by a graph-based

message-passing network [13]:

$$S_t = GCL(H_t, A), \tag{10}$$

where A is the adjacency matrix of the corresponding network. By applying GCL to each time-aware representation  $H_t$ , we obtain the refined traffic representations  $(S_{t-T_1+1}, \ldots, S_t)$ .

Using TCL and GCL, we construct TSRL by two "sandwich" structures, *i.e.*, TCL  $\rightarrow$  GCL  $\rightarrow$  TCL. To facilitate understanding, we provide a visual aid in Fig. 3(b) that aims to explain the "sandwich" concept. The final output of our TSRL is a representation  $\mathcal{Z} \in \mathbb{R}^{T' \times N \times D}$  with the temporal dimension T':

$$\mathcal{Z} = (\mathbf{Z}_{t-T'+1}, \dots, \mathbf{Z}_t) = TSRL(\mathcal{X}, \mathbf{A}). \tag{11}$$

Since designing the TSRL module is not the focus of this paper, we adopted the components of a classic architecture from the ST domain (in particular STGCN [7]) as the backbone, with a trade-off between performance and efficiency.

2) Contextual Disentanglement: The representation  $\mathcal{Z}$  involves information about invariant and variant ST contexts. To disentangle these two types of information, we propose to use two TSRL modules to generate  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$ :

$$\mathcal{Z}_I = \text{TSRL}_1(\mathcal{X}, \mathbf{A}), \quad \mathcal{Z}_V = \text{TSRL}_2(\mathcal{X}, \mathbf{A}),$$
 (12)

where  $\mathcal{Z}_I, \mathcal{Z}_V \in \mathbb{R}^{T' \times N \times D}$ . In Theorem 1, DCA requires the independence of invariant and variant ST contexts. To meet such requirement, we disentangle  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  using a mutual information (MI) minimizing loss as

$$\arg\min_{\mathcal{Z}_I,\mathcal{Z}_V} -\mathbb{E}_{p(\mathcal{Z}_I,\mathcal{Z}_V)}[\log p(\mathcal{Z}_I) + \log p(\mathcal{Z}_V) - \log p(\mathcal{Z}_I,\mathcal{Z}_V)]. \tag{13}$$

Because the marginal and joint distributions of  $\mathcal{Z}_I$ ,  $\mathcal{Z}_V$  are unknown, we adopt vCLUB [14] to approximate the loss function in Eq. (13) by

$$\mathcal{L}_{D} = \frac{1}{M} \sum_{i=1}^{M} \left[ \log q_{\theta} \left( \mathcal{Z}_{I}^{(i)} | \mathcal{Z}_{V}^{(i)} \right) - \frac{1}{M} \sum_{j=1}^{M} \left[ \log q_{\theta} \left( \mathcal{Z}_{I}^{(j)} | \mathcal{Z}_{V}^{(i)} \right) \right] \right], \tag{14}$$

where M is the sample size.  $q_{\theta}(\mathcal{Z}_{I}|\mathcal{Z}_{V})$  is the variational distribution, which is estimated by  $\mathcal{N}(\mathcal{Z}_{I}|\mu(\mathcal{Z}_{V}),\sigma^{2}(\mathcal{Z}_{V}))$  with the reparameterization technique [15].  $\mu(\cdot)$  and  $\sigma(\cdot)$  are implemented by a two-layer MLP.

# C. Context-Oriented Self-Supervised Deconfouning

Although we disentangle  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  in Sec. III-B2, the correspondence between  $\mathcal{Z}_I$ ,  $\mathcal{Z}_V$  and invariant/variant ST context is still not determined. To orient  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  towards invariant and variant ST contexts, we propose a context-oriented self-supervised deconfounding (CO-SSD) module. Specifically, i) we devise three self-supervised tasks to inject context information into  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$ , ii) we utilize an adversarial learning module to fuse variant context information into  $\mathcal{Z}_V$  and exclude such information from  $\mathcal{Z}_I$ .

1) Self-Supervised Tasks: Since not all ST context data is available, we use some representative ST contexts as self-supervised signals to inject context information into representations  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$ . Specifically, we categorize ST contexts into three classes from conceptually different perspectives, *i.e.*, temporal, spatial, and semantic, based on the unique properties of ST data. We then carefully select representative and easily collected contexts from each class, such as temporal index, spatial location, and traffic capacity. These selected contexts serve as self-supervised signals that can instruct our model to effectively identify more latent ST context variables. By designing self-supervised tasks that capitalize on these signals, we ensure that our model can learn robust representations capable of accommodating previously unseen ST contexts.

Task #1: Spatial Location Classification. The spatial location of a region is reflective of its surrounding ST contexts. They may vary with different locations, thereby changing the dependency of past data and future data (e.g.,  $(x_{t-T+1}, \ldots, x_t) \to x_{t+1}$ ). For example, such dependency in a transportation hub can significantly differ from that in a residential area. Therefore, we propose a spatial location perception task to preserve the ST contexts of each region.

Firstly, for node (region)  $v_n \in \mathcal{V}$ , we utilize the node ID to assign it a unique one-hot location label,  $y_n^{(1)} \in \mathbb{R}^N$ . We then optimize the *spatial location perception* task by the crossentropy loss as

$$L_{sl}(\mathcal{Z}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{N} y_{n,m}^{(1)} \log \left( \hat{y}_{n,m}^{(1)} \right),$$
  
s. t.  $\hat{y}_{n}^{(1)} = \text{SoftMax}(g_{1}(\tilde{z}_{n})),$  (15)

where  $\hat{y}_n^{(1)} \in \mathbb{R}^N$  is the predicted location label vector with items  $\hat{y}_{n,m}$ , and  $g_1(\cdot)$  is implemented by a two-layer MLP.  $\tilde{z}_n \in \mathbb{R}^D$  is a node representation generated by a 1-D convolution network:  $\tilde{z}_n = \text{Conv1D}(z_{t-T'+1,n},\ldots,z_{t,n})$ . The input of the Conv1D network is rows of  $(Z_{t-T'+1,n},\ldots,Z_{t,n})$ .

Task #2: Temporal Index Identification. Time-varying ST contexts, such as holidays and weather, often affect traffic data distributions. For example, holidays can flatten the curve of the evening peak. This produces a significantly different data distribution from the normal evening peak in workdays.

To utilize the temporal index, we propose a temporal index classification task.

Specifically, we divide a day into 24 time slots, each of which is a category. To distinguish workdays and holidays, we use different indexes, resulting in K=48 temporal indexes in total. For a given traffic sample  $(\mathcal{X}, \mathbf{Y})$ , we use the temporal index of  $\mathbf{Y}$  as the ground truth. We denote the one-hot temporal index as  $\mathbf{y}^{(2)} \in \mathbb{R}^K$ . The optimization objective of the temporal index classification task is

$$L_{ti}(\mathcal{Z}) = \sum_{k=1}^{K} y_k^{(2)} \log \left( \hat{y}_k^{(2)} \right),$$
s. t.  $\hat{\boldsymbol{y}}^{(2)} = \operatorname{SoftMax} \left( \frac{1}{N} \sum_{n=1}^{N} (g_2(\tilde{\boldsymbol{z}}_n)) \right),$  (16)

where  $\hat{y}^{(2)} \in \mathbb{R}^K$  is the predicted temporal index vector with items  $\hat{y}_k^{(2)}$ .  $g_2$  is a two-layer MLP used to refine the n-th node representation  $\tilde{z}_n$ .

Task #3: Traffic Load Prediction. The traffic load is a kind of important semantic context that describes the congestion level of a road segment or a spatial region. It also has an impact on the change of future traffic. For example, when the traffic load reaches saturation, the traffic is more likely to be congested and traffic flow may drop in the near future. Therefore, we propose a traffic load prediction task to enable the traffic representation to be aware of the current traffic state.

Specifically, we approximate the traffic load capacity of the n-th node by using the historical maximum traffic flow, i.e.,  $CP_n = \max(\boldsymbol{x}_{t,n}) \in \mathbb{R}^d, t \in [1,\tau].$   $\tau$  is the total number of time steps in the training set.  $\max(\cdot)$  keeps the original dimension of the input. Then, we divide traffic flows into 6 load levels and calculate the traffic load of the n-th node by  $\boldsymbol{y}_n^{(3)} = \lceil 5\boldsymbol{x}_{t+1,n}/CP_n \rceil \in \{0,\dots,5\}^d$ . Since there may be some missing load states and the load states are quite imbalanced in practice, we adopt the mean square error (MSE) loss to optimize the *traffic load prediction* task as

$$L_{tl}(\mathcal{Z}) = \frac{1}{N} \sum_{n=1}^{N} \left\| g_3(\tilde{z}_n) - y_n^{(3)} \right\|^2,$$
 (17)

where  $g_3(\cdot)$  is the traffic load predictor implemented by a two-layer MLP, and  $\tilde{z}_i$  is the same node representation as in Task #1.

2) Adversarial Learning: We use the above self-supervised losses in Eq. (15), (16) and (17) to train representation  $\mathcal{Z}_V$ , making it involve information from variant ST context. Furthermore, we expect  $\mathcal{Z}_I$  not to involve such information. To this end, we introduce the gradient reversal layer (GRL) [16] to exclude variant context information from  $\mathcal{Z}_I$ . Specifically, we use the same losses as  $\mathcal{Z}_V$ , but in the back-propagation process, we multiply the gradients of  $\mathcal{Z}_I$  by a negative factor  $-\eta$  which reverses the gradient direction. This drives the representation learning network of  $\mathcal{Z}_I$  away from the optimization direction of the self-supervised task. Note the factor  $\eta$  equals 1 in this paper. By defining  $\bar{\mathcal{Z}}_I = \text{GRL}(\mathcal{Z}_I)$ , we have the loss of CO-SSD

# Algorithm 1 Training Algorithm of STEVE

**Input:** Traffic network  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ , historical traffic data  $\mathcal{X} = (\mathbf{X}_1, \dots, \mathbf{X}_{\tau})$ , and hyperparameters.

Output: The learned model parameters.

- 1: for doavailable  $t \in \{1, \dots, \tau\}$
- 2:  $X \leftarrow \{X_{t-T+1}, \dots, X_t; A\}$ .  $\triangleright$  Input data
- 3:  $Y \leftarrow X_{t+1}$ . 4: Put  $\{X, Y\}$  into  $\mathcal{D}_{train}$ .
- 5: Initialize all trainable parameters.
- 6: while stopping criterion is not met do
- 7: Randomly select a batch  $\mathcal{D}_{batch}$  from  $\mathcal{D}_{train}$ .
- 8: Use  $\mathcal{D}_{batch}$  to compute  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  via Eq. (12).
- 9: Use  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  to compute  $\mathcal{L}_D$  via Eq. (14).
- 10: Use  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  to compute  $\mathcal{L}_S$  via Eq. (18).
- 11: Use  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  to compute  $\mathcal{L}_P$  via Eq. (19).
- 12:  $\mathcal{L}_O = \mathcal{L}_P + \mathcal{L}_S + \mathcal{L}_D$ .
- 13: Use  $\mathcal{L}_O$  to compute gradients of all parameters via the backpropagation algorithm.
- 14: **for** parameter  $\theta$  in STEVE **do**
- 15:  $\theta = \theta \eta \cdot \nabla_{\theta} \mathcal{L}_{O}$

 $\triangleright \eta$  is learning rate

▶ Label

16: return Parameters of STEVE.

#### module as

$$\mathcal{L}_S = L_{sl}(\mathcal{Z}_V) + L_{ti}(\mathcal{Z}_V) + L_{tl}(\mathcal{Z}_V) + L_{sl}(\bar{\mathcal{Z}}_I) + L_{ti}(\bar{\mathcal{Z}}_I) + L_{tl}(\bar{\mathcal{Z}}_I).$$
(18)

In this equation, we simply minimize the self-supervised losses of  $\mathcal{Z}_V$  to make it perform well on these tasks. Meanwhile, we use GRL to induce  $\mathcal{Z}_I$  to perform poorly on these tasks. The training process of  $\mathcal{Z}_V$  and  $\mathcal{Z}_I$  is like a min-max game, so we call it adversarial learning. As a result, the adversarial learning loss in Eq. (18) fuses the variant context information into  $\mathcal{Z}_V$  and simultaneously excludes such information from  $\mathcal{Z}_I$ .

# D. Model Training

In the learning process of STEVE, we fed the traffic prediction  $\hat{Y}$  in Eq. (8) into the following loss:

$$\mathcal{L}_{P} = \frac{1}{N * F} \sum_{i=1}^{N} \sum_{j=1}^{F} |y_{i,j} - \hat{y}_{i,j}|, \qquad (19)$$

where N is the number of nodes, and F is the dimensionality of the output.  $\hat{y}_{i,j}$  is the element of  $\hat{Y}$ , and  $y_{i,j}$  denotes the ground truth of future traffic state. Finally, we obtain the overall loss by incorporating the independence regularization in Eq. (14) and self-supervised adversarial loss in Eq. (18) into the joint learning objective:

$$\mathcal{L}_O = \mathcal{L}_P + \mathcal{L}_S + \mathcal{L}_D. \tag{20}$$

This learning objective derived from Eq. (4) can remove spurious correlations latent in ST traffic data, and enable the proposed model with better robustness in OOD scenarios.

Our model can be trained end-to-end via the backpropagation algorithm. The entire training procedure is summarized into Algo. 1. In lines 1-4, we construct training data. In lines 6-15, we iteratively optimize STEVE by gradient descent until the stopping criterion is met. Specifically, in lines 7-13, we first select a random batch of data and then apply the forward-backward operation on the whole model to get gradients of all

TABLE II STATISTICS OF DATASETS.

Dataset	NYCBike1	NYCBike2	NYCTaxi	BJTaxi
Data type	Bike	rental	Taxi (	GPS
Time interval	1 hour	30 min	30 min	30 min
# regions	16×8	$10\times20$	$10 \times 20$	$32\times32$
# taxis/bikes	6.8k+	2.6m+	22m+	34k+
# seq length	4392	2880	2880	5596

parameters. At last, in lines 14-15, we update all parameters within STEVE by gradient descent.

#### IV. EXPERIMENTS

In this section, we compare our STEVE against a diverse set of ST traffic forecasting approaches in temporal and spatial OOD settings, and report the results of a detailed empirical analysis of STEVE.

# A. Datasets and Experimental Settings

1) Datasets: We conducted experiments on four commonly used real-world large-scale datasets released by [13]. These datasets are generated by millions of taxis or bikes on average and contain thousands of time steps and hundreds of regions. The statistical information is summarized in Tab. II. Two of them are bike datasets, while the others are taxi datasets. Bike data record bike rental demands. Taxi data record the number of taxis coming to and departing from a region given a specific time interval, *i.e.*, inflow and outflow.

We give more detailed descriptions of the four datasets as follows. NYCBike series datasets consist of one hourly level dataset from 1/Apr/2014 to 30/Sept/2014 (NYCBike1 [17]) and one 30-minute level dataset from 1/Jul/2016 to 29/Aug/2016 (NYCBike2 [18]). NYCTaxi [18] measures the 30-minute level taxi flow from 1/Jan/2015 to 01/Mar/2015. BJTaxi [17] is also a 30-minute level taxi dataset from 01/Mar/2015 to 30/Jun/2015, collected in Beijing city. For all datasets, the traffic network is constructed by the adjacency relation of regions. We use previous 4-hour traffic flows and past 3-day flows around the predicted time as input. This can facilitate the modeling of shifted temporal correlations [18]. We adopt a sliding window strategy to generate samples, and then split each dataset into the training, validation, and test sets with a ratio of 7:1:2.

2) Baselines and Metrics: Since traditional statistical models and shallow machine learning methods have proven difficult to effectively model ST traffic data [1], [10], we compare STEVE with recent state-of-the-art baselines as follows.

# i) Graph-Based Spatial-Temporal Methods:

- **STGCN** [7]: a graph convolution-based model that combines 1D-convolution to capture spatial and temporal correlations.
- AGCRN [1]: it enhances the classical graph convolution with an adaptive adjacency matrix and combines it into RNN to model ST data.
- **ST-Norm** [19]: it introduces temporal normalization and spatial normalization modules to refine the high-frequency and local components of the original ST data, respectively.

# ii) Series- and Graph-based OOD Approaches:

• AdaRNN [20]: a time series model that addresses distribution

TABLE III
TEMPORAL OOD RESULTS ON FOUR DATASETS w.r.t. MAE AND MAPE (%). WE REPORT THE AVERAGE RESULT OF THREE RUNS WITH THE BEST IN BOLD.
THE ROW TITLE ON EACH DATASET INDICATES A TEST SCENARIO. ROW AVG. MEANS THE AVERAGE RESULT OF ALL TEST SCENARIOS.

	Method	STO	GCN	AG	CRN	ST-I	Norm	Ada	RNN	CC	OST	CI	GA	STN	ISCM	Ca	uST	ST	EVE
Dataset	Metric	MAE	MAPE																
ke1	Workday	5.50	25.28	5.44	25.19	5.46	25.46	7.22	29.64	6.64	29.67	6.47	29.27	5.97	26.67	8.01	29.86	5.18	22.63
CB:	Holiday	5.16	29.98	5.06	29.71	5.48	26.45	6.13	33.34	5.76	32.61	5.29	29.91	6.34	37.62	5.67	29.53	4.87	26.17
NYCBike1	Avg.	5.33	27.63	5.25	27.45	5.47	25.96	6.68	31.49	6.49	33.65	6.12	30.94	5.63	28.29	6.84	29.70	5.03	24.40
NYCBike2	Workday	5.43	25.09	5.35	24.62	5.57	26.25	8.18	36.54	7.06	31.23	6.05	31.49	6.15	27.88	7.26	28.87	4.82	20.54
CBi	Holiday	5.53	30.71	5.43	30.15	5.39	27.62	7.35	28.47	7.50	39.32	5.86	28.45	5.76	31.13	5.50	28.72	4.88	24.61
NX	Avg.	5.48	27.90	5.39	27.39	5.48	26.94	5.96	29.97	7.28	35.28	7.76	32.51	5.96	29.51	6.38	28.80	4.85	22.58
axi	Workday	11.38	18.90	10.87	18.28	16.57	31.47	15.16	41.49	13.14	32.80	15.23	18.95	14.69	23.63	16.08	31.98	10.53	16.72
NYCTaxi	Holiday	11.32	18.69	10.91	17.99	17.13	30.55	16.96	32.21	13.02	30.37	15.34	21.51	14.95	23.39	15.61	30.22	10.58	16.25
E	Avg.	11.35	18.80	10.89	18.14	16.85	31.01	16.06	36.85	13.08	31.59	15.29	20.23	14.82	23.51	15.85	31.10	10.56	16.49
.ix	Workday	12.52	14.91	11.99	14.68	13.26	16.75	19.63	21.89	14.05	17.10	13.47	16.71	13.80	16.96	17.31	19.27	11.68	14.20
BJTaxi	Holiday	11.77	19.34	11.11	18.92	13.36	18.27	17.78	28.79	13.87	22.41	12.69	22.24	11.29	19.36	25.93	30.15	11.01	18.90
	Avg.	12.14	17.13	11.55	16.80	13.31	17.51	18.75	25.34	13.96	19.76	13.08	19.48	12.55	18.16	21.62	24.71	11.34	16.55
Co	ount		0		0		1		0		0		0		0		0		23

shift challenges. It clusters historical time sequences into different classes and dynamically matches input data to these classes to identify contextual information.

- **COST** [21]: a time series model that disentangles season and trend information from a causal lens to enhance model robustness. The backbone is temporal convolution networks.
- CIGA [22]: it is a graph model that captures the invariance of graphs via causal models to guarantee OOD generalization under various distribution shifts.

# iii) Spatio-Temporal OOD Models:

- STNSCM [23]: it is a spatio-temporal model that neuralizes a structural causal model and incorporates external conditions such as time factors and weather for traffic prediction in OOD scenarios. For fair comparisons, we use time factors as the only external conditions.
- CauST [24]: it is a spatio-temporal model that captures invariant relations for OOD generalization.

To evaluate the forecasting performance of different methods, we use two common metrics: Mean Average Error (MAE) and Mean Average Percentage Error (MAPE).

3) Implementation Details of STEVE: Our STEVE is implemented with PyTorch. Both the temporal and spatial convolution kernel sizes in TSRL are searched in  $\{2,3,4,5\}$  and the optimal setting is 3 for all datasets. The number of "sandwich" layers is searched in  $\{1,2,3,4\}$  and the best setting is 2 for all datasets. We also conduct a grid search for the representation dimension among  $\{16,32,64,128\}$ . Ultimately, for the BJTaxi and NYCBike2 datasets, we set the representation dimension to 32, while for the NYCBike1 and NYCTaxi datasets, it is set to 64. We optimize our STEVE with the Adam optimizer and the initial learning is set to 0.001. We utilize a dynamic weight-averaging strategy [25] to balance the learning rate between multiple self-supervised tasks. For any more details, please refer to our code at https://github.com/ShotDownDiane/STEVE.

# B. Performance Comparison

- 1) Settings: The commonly used evaluation for ST traffic forecasting mixes up different temporal and spatial scenarios. However, some real-world model users may be concerned about accurate forecasting results in particular scenarios, e.g., holiday time or suburban areas. Since the data distribution in particular scenarios usually differs from the mixture distribution, it requires the generalization ability of models for **distribution shifts** in test data, i.e., OOD data. To emulate distribution shifts and assess models' OOD generalization, we partition the test data into distinct scenarios for individual evaluation. For example, when training, the data comprise both workday and holiday samples (usually in the ratio of 5:2). During the testing phase, we deliberately structure the test data to solely consist of either workday samples or holiday samples. That is, we shift the test ratio to 1:0 or 0:1 to mirror practical scenarios.
- 2) Results: Next, we use the above settings to construct temporal and spatial OOD settings for evaluation. The results are shown in Tab. III and Tab. IV, respectively.

**Temporal OOD Forecasting.** We split the test data into workdays and holidays and shift the data ratio from roughly 5:2 (in the training set) to 1:0 and 0:1 (in the test set). We then test our STEVE and all baselines on both OOD scenarios. From Tab. III, we can observe that: i) The proposed STEVE significantly improves the forecasting performance (winning counts in the last row) across all datasets. ii) The STEVE completely beats its canonical degradation STGCN, which supports the confounding assumption of ST context C. This also indicates the necessity of removing spurious correlations of X and Y (caused by confounder C) and incorporating ST context information into ST dependency modeling. iii) Our proposed STEVE outperforms recent OOD generalization-related models, such as series-based AdaRNN and COST, graph-

_	Method	STO	GCN	AG	CRN	ST-I	Norm	Ada	RNN	CC	OST	CI	GA	STN	ISCM	Ca	uST	ST	EVE
Dataset	Metric	MAE	MAPE	MAE	MAPE														
	c0	2.96	33.86	2.93	33.46	3.40	33.29	3.27	30.80	3.13	31.43	3.21	34.82	3.03	26.84	4.07	38.46	2.23	23.90
e-1	c1	4.36	28.50	4.35	28.88	4.42	27.42	5.45	32.83	4.98	32.16	4.73	28.69	4.60	27.00	6.00	32.83	4.20	26.80
NYCBike1	c2	5.81	24.56	5.87	25.41	5.76	24.31	8.00	31.57	7.28	33.65	6.72	25.08	5.86	23.28	7.97	27.33	5.63	23.23
Σ	c3	7.53	22.28	7.56	22.61	7.98	21.26	10.35	28.52	9.24	29.67	9.07	21.99	8.37	20.24	9.61	24.12	7.24	20.69
	Avg.	5.16	27.30	5.18	27.59	5.39	26.57	6.77	30.93	6.16	31.73	5.93	27.64	5.47	24.34	6.91	30.69	4.83	23.66
23	c0	3.94	37.47	3.80	36.73	3.46	33.33	5.02	39.72	3.91	36.68	3.31	37.79	6.11	29.68	4.30	38.22	2.71	24.04
NYCBike2	c1	4.90	26.47	4.74	26.04	5.04	28.86	7.56	38.49	5.87	31.92	5.64	28.77	6.71	27.62	6.46	30.75	4.49	23.57
Ç	c2	7.05	20.17	7.00	19.75	7.01	21.42	13.53	40.66	10.38	33.54	9.56	29.34	5.82	27.68	10.78	25.85	<u>6.71</u>	18.70
É	Avg.	5.30	28.04	5.18	27.51	5.17	27.87	8.71	39.62	6.72	34.05	6.17	31.97	6.21	28.33	7.18	31.61	4.64	22.10
	c0	3.97	27.68	3.77	26.14	6.01	45.77	4.81	35.52	4.70	31.11	5.32	27.11	4.75	24.11	6.95	42.07	3.50	22.46
iXi	c1	8.31	16.80	8.17	16.76	13.09	30.53	17.97	35.59	15.10	33.55	8.76	17.02	8.54	22.65	15.85	29.67	8.06	16.36
NYCTaxi	c2	17.17	11.42	17.12	11.34	23.92	18.23	29.20	40.89	38.92	34.19	19.19	13.33	17.72	12.43	36.19	22.23	16.62	11.14
Σ	c3	27.56	9.74	27.68	9.65	39.05	16.37	35.25	46.73	67.02	33.07	27.65	16.09	24.29	11.11	65.90	19.71	25.96	9.16
	Avg.	14.25	16.41	14.19	15.97	20.51	27.72	21.81	39.65	31.44	32.98	15.23	18.39	13.83	17.58	31.22	28.42	13.53	14.78
	c0	4.97	23.88	5.93	28.87	5.79	25.63	6.33	27.86	5.18	23.76	7.81	29.14	6.79	23.87	6.35	28.44	4.78	22.87
	c1	9.45	15.41	11.16	15.37	10.71	16.87	13.99	24.17	10.35	18.62	12.09	16.88	10.87	18.43	13.51	20.77	9.23	15.30
axi	c2	13.84	12.38	20.51	12.77	15.50	13.38	22.63	22.03	15.75	15.91	16.29	13.91	14.09	14.22	22.75	19.69	13.39	12.10
BJTaxi	c3	20.64	10.56	21.71	11.46	22.19	10.79	33.67	20.02	24.80	14.50	22.01	12.99	21.96	12.22	37.88	18.95	19.71	10.18
H	c4	30.13	9.34	31.44	9.29	31.08	8.99	52.23	17.50	37.31	12.94	31.03	10.58	29.23	9.64	58.99	18.07	27.88	8.78
	Avg.	15.80	14.31	18.15	15.55	17.05	15.13	25.77	22.32	18.68	17.15	17.85	16.70	16.59	15.67	27.90	21.18	15.00	13.85
Co	ount		0		0		0		0		0		0		3		0	3	37

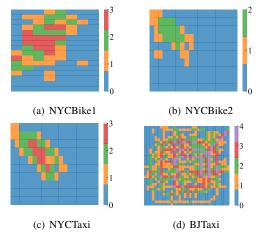


Fig. 4. Spatial clustering results of all datasets. The cluster identification (ID) is next to the color bar. A larger cluster ID means a higher level of popularity.

based CIGA, and ST-based STNSCM and CauST. The MAE decreases 32.3%, 23.5%, 20.1%, 16.7%, and 31.8% on average. Series-based and graph-based OOD methods overlook the spatial and temporal dependency modeling respectively, leading to their poor performance. STNSCM delivers unsatisfactory results, demonstrating its vulnerability when external data is not fully accessible. *iv*) Interestingly, our STEVE largely surpasses CauST, which primarily focuses on invariant learning for OOD generalization. This underscores the significance of considering both invariant and variant relations in OOD ST forecasting.

**Spatial OOD Forecasting.** In the spatial scenario, we split all regions into several clusters to simulate urban functional areas. Since there is no function label, we use k-means clustering algorithm to label the regions. The best k is determined by the Silhouette Coefficient metric [26]. The input of k-means is (mean, median) of each region's historical traffic flows. Fig. 4 presents the clustering results of all datasets, which exhibit some meaningful patterns. Taking BJTaxi as an example, the clustering results imply the suburbs (ID 0) and ring roads (ID 3). Tab. IV presents the performance comparison, from which we can observe that: i) Our STEVE greatly outperforms other methods, and the findings i, ii, and iv in the temporal OOD settings still hold for the spatial OOD scenarios. ii) The proposed method shows better results than recent OODrelated AdaRNN, COST, CIGA, STNSCM, and CauST on MAE by decreasing 37.2%, 28.87%, 18.9%, 13.5%, and 40.72% on average. On the NYCBike1, NYCBike2, and NYCTaxi datasets, STNSCM performs better in the popular areas (c3, c2, and c3), and our method surpasses it in other non-popular areas. We attribute this to a specific example, in which the effectiveness of prediction capacity is reflected with model robustness, especially in the case of sparse data.

**Significance Test.** To further emphasize the substantial improvement of our STEVE over the baseline models, we draw the critical difference (CD) diagram to conduct a Nemenyi significance test. As shown in Fig. 5, we can observe that our

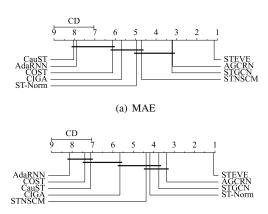


Fig. 5. Critical difference (CD) diagram of the Nemenyi test w.r.t. metrics MAE and MAPE. The horizontal axis depicts the average ranking of each model across all scenarios, with lower rankings indicating superior performance. Bold black lines connect two models when the difference in their average rankings is below the CD value (at a 5% significance level), indicating statistical insignificance. Otherwise, the two models are statistically significantly different.

(b) MAPE

 $\label{eq:table_variable} TABLE\ V$  Ablation study of STEVE on the average performance.

	Dataset	NYC	Bike1	NYC	Bike2	NYO	CTaxi	BJ	Taxi
	Metric	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
$\overline{}$	STEVE	5.03	24.40	4.85	22.58	10.56	16.49	11.34	16.55
00D	w/o cd	5.05	24.58	4.86	23.16	10.74	17.18	11.37	16.68
	w/o gr	5.07	24.76	4.86	23.46	10.60	17.23	11.42	16.97
ora	w/o idp	5.21	25.77	4.98	24.61	11.15	18.10	12.17	17.13
Temporal	w/o sl	5.07	24.87	4.89	23.18	10.67	16.93	11.38	16.71
F	w/o ti	5.08	25.38	4.89	23.47	11.04	17.36	11.43	16.89
	w/o tl	5.15	25.36	4.90	22.67	10.61	17.42	11.48	16.73
	STEVE	4.83	23.66	4.64	22.10	13.53	14.78	15.00	13.85
00D	w/o cd	4.87	24.30	4.71	22.43	13.58	15.45	15.17	13.87
	w/o gr	4.92	23.97	4.74	22.81	13.71	15.29	15.34	14.16
Spatial	w/o idp	5.05	24.87	4.98	24.15	15.01	15.81	16.10	15.13
pat	w/o sl	4.89	23.93	4.69	22.44	14.22	15.61	15.15	13.94
S	w/o ti	4.89	24.41	4.66	22.45	13.98	15.66	15.09	13.86
	w/o tl	4.84	23.85	4.70	22.23	13.61	14.79	15.04	13.85

STEVE outperforms the best baseline significantly at a 5% significance level.

# C. Ablation Study

In this part, we carry out ablation experiments from two aspects to verify our model design: the important components and the backbone architecture of the TSRL module.

**Ablation of important components.** We design six variants to test the effectiveness of STEVE's components: i) w/o cd removes the contextual disentanglement component by disabling the mutual information ii) w/o gr removes the gradient reversal layer in Eq. (18). iii) w/o idp is a combination of w/o cd and w/o gr, violating the independence requirement of DCA. The next three variants are about self-supervised tasks. iv) w/o sl removes the spatial location classification task. v) w/o tl removes the temporal index identification task. vi) w/o tl removes the traffic load prediction task.

Tab. V presents the results of our STEVE and its six variants. From the results, we can observe that: *i*) The variant *w/o idp* 

TABLE VI ABLATION STUDY OF BACKBONE ARCHITECTURE IN TSRL.

Variants [# Parameters]	Temporal OOD   Spatial OOD								
	MAE	MAPE	MAE	MAPE					
STEVE-AGCRN [2486k] STEVE [282k]	5.00	25.10 24.40	4.82	23.72					
STEVE [282k]	5.05	24.40	4.83	23.66					

performs worse than the STEVE with a large margin. This indicates that maintaining a strong disentanglement between  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  can satisfy the independence requirement of DCA, thus eliminating spurious correlations that impair OOD generalization. However, only w/o gr or w/o cd does not seriously degrade performance, which suggests that contextual disentanglement and gradient reversal layer are complementary in decoupling  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$ . ii) The results of w/o sl, w/o ti, and w/o tl suggest that every self-supervised task plays a crucial role in improving OOD performance. Intuitively, in temporal OOD settings, the tasks of temporal index identification and traffic load prediction contribute more than the spatial location classification task. However, in spatial OOD settings, the spatial location classification task becomes the most useful auxiliary task. In summary, each designed component has a positive effect on the performance improvement of our STEVE.

Ablation of backbone architecture. The backbone architecture of STEVE, *i.e.*, TSRL, aims to encode traffic sequence data as traffic representations. As TSRL is designed as a loosely coupled module, there can be many instances with different choices of implementation. In this paper, we made a trade-off between performance and efficiency and adopted a classical architecture from the ST domain, in particular STGCN [7], as the backbone in our TSRL module. To verify it, we design a variant called STEVE-AGCRN that replaces STGCN with the best baseline AGCRN [1]. The experiment results on NYCBike1 are shown in Tab. VI. Our STEVE achieves similar performance to STEVE-AGCRN with only one-tenth the number of parameters, which indicates the superiority of the current backbone selection.

#### D. Parameter Sensitivity

In this part, we conduct experiments to analyze the impacts of critical hyper-parameters: spatial kernel size, temporal kernel size, number of "sandwich" layers, and hidden dimension.

In Fig. 6, we present the ST forecasting results over all datasets with different parameters. Firstly, the effect of spatial and temporal kernel size is shown in Fig. 6(a), where we vary them from 2 to 5 individually. We can see that 3 is the optimal setting for both kernel sizes. This verifies the spatial and temporal localized characteristics of traffic data. Secondly, the effect of "sandwich" layer number is shown in Fig. 6(b), which demonstrates that a shallow-layer encoder is insufficient to encode spatial and temporal information, while a deep-layer encoder suffers from the over-smoothing issue of graph convolution and exhibits a performance drop. Thirdly, the effect of hidden dimension is given in Fig. 6(c), where we vary it in the set {16, 32, 64, 128}. The results indicate 64 as

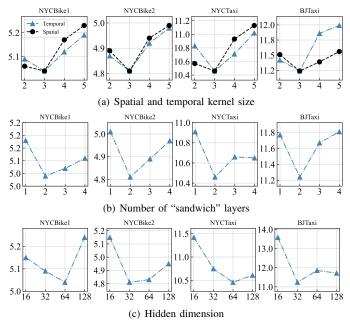


Fig. 6. Parameter sensitivity of STEVE using MAE metric.

the optimal settings for NYCBike1 and NYCTaxi datasets and 32 for NYCBike2 and BJTaxi. Since different datasets have different spatio-temporal dependencies, it is reasonable to use different hidden dimensions for them.

# E. Case Study

Effectiveness of Partial ST Contexts. Because the complete context information is unknown, we use partial context information in Sec. III-C to cover common ST contexts and inject such information into representations. We here utilize an external context, which is unseen in training data, to evaluate the effectiveness of partial ST contexts. Specifically, we first collect weather data for traffic samples in BJTaxi, resulting in 6 different types of weather. The ratio in existing training data is 48:29:7:7:5:4. For test data, we randomly select traffic samples of one type of weather for testing. This makes the ratio 0:0:1:0:0:0 (for example), leading to a distribution shift in test data. Fig. 7(a) presents the test results of traffic forecasting in each type of weather. We can observe that our STEVE consistently performs better than other OOD-related baselines, especially in unusual and extreme weather such as "sprinkle". The reason is that STEVE manages to learn the latent distribution of weather contexts that is unseen in the training process by using partial ST contexts.

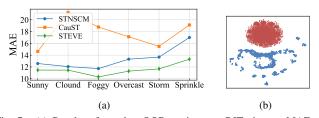


Fig. 7. (a) Results of weather OOD settings on BJTaxi w.r.t. MAE. (b) Representation visualization of  $\mathcal{Z}_I$  (in red) and  $\mathcal{Z}_V$  (in blue) on BJTaxi.

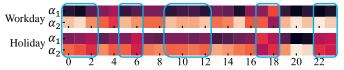


Fig. 8. Visualization of the learned priors. The horizontal axis represents the time of day. A brighter pixel means a larger value. All values lie in [0, 1].

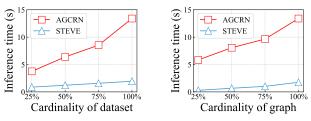


Fig. 9. Scalability performance vs. cardinality

**Visualization of Representation.** Traffic representations  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  are supposed to be disentangled and aligned to invariant and variant ST contexts, respectively. To explore whether our model could learn meaningful representations, we visualize  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  of BJTaxi by t-SNE [27]. As shown in Fig. 7(b),  $\mathcal{Z}_I$  and  $\mathcal{Z}_V$  are separated into two parts, indicating the success of disentanglement. Moreover, the phenomenon that  $\mathcal{Z}_I$  are more compact and  $\mathcal{Z}_V$  are scattered in a large area, suggests they have encoded distinctive ST context information.

**Adaptation to Distribution Shifts.** Since priors of invariant and variant ST contexts, *i.e.*,  $\alpha_1, \alpha_2$ , are unknown, we produce them in a learnable manner. To verify the effectiveness of the learned priors, we visualize them in Fig. 8. Distinct priors for workdays and holidays indicate their adaptation to distribution shifts, such as the evening peaks marked in the rectangle.

# F. Scalability and Efficiency

**Model Scalability.** In the sequel, we explore the scalability performance of STEVE compared with AGCRN (the best baseline), focusing on their ability to handle variations in dataset size and graph size. The evaluation employs the BJTaxi dataset that contains traffic data from 1024 graph nodes over a 4-month period. Fig. 9 depicts the experimental results. Regarding the dataset size, 25% denotes a one-month dataset, 50% denotes a two-month dataset, and so on. For the graph size, we decompose the input graph into four connected subgraphs with the same node number. Here, 25% implies using nodes from the first subgraph to extract an adjacency matrix from the original one, 50% involves nodes from the first two subgraphs, and so on. The first observation is that the prediction time increases with the expansion of cardinality in both dataset and graph size. Second, STEVE exhibits a significant improvement in prediction efficiency compared to AGCRN. Notably, the prediction time of STEVE remains more stable with cardinality growth in both scenarios, whereas AGCRN experiences a rapid increase. The disparity arises from AGCRN's reliance on an RNN structure to capture temporal dependencies, leading to accumulated time costs, especially with larger datasets. In

TABLE VII
TIME COMPLEXITY OF KEY COMPONENTS OF STEVE.

Components	Time Complexity
Traffic sequence representation learning (TSRL) Contextual disentanglement (CD) Context-oriented self-supervised learning (CO-SSD)	$ \begin{array}{ c c } \hline O(NT + TN^2) \\ O(NTM) \\ O(NT + N) \\ \hline \end{array} $

TABLE VIII EFFICIENCY EVALUATION BY TRAINING/INFERENCE TIME PER EPOCH (S).

Methods	NYCBike1	NYCBike2	NYCTaxi	BJTaxi
	24.23/2.52	25.02/1.98	20.91/3.13	221.41/13.40
	6.46/0.73	8.76/0.71	9.92/0.89	56.45/1.95

contrast, STEVE utilizes a convolutional structure that is more efficient than RNN. On the other hand, the prediction time of STEVE is more stable as the number of graph nodes increases, which is not the case for AGCRN. This is attributed to AGCRN requiring learning an adaptive adjacency matrix, incurring a quadratic time cost with growing graph size. In contrast, STEVE adopts the original adjacency matrix, avoiding additional time costs. Overall, the STEVE demonstrates good potential scalability in large-scale ST forecasting.

**Model Efficiency.** In this part, we investigate the efficiency of our STEVE both theoretically and practically. As presented in Tab. VII, we conduct a time complexity analysis on key components of STEVE. The symbols are consistent with Tab. I.

- TSRL: Within the traffic sequence representation learning module, there exist two temporal convolution layers (TCL) and one graph convolutional layer (GCL). The time complexity of the TCL is denoted as  $O(2N((T-K_t+1)K_t*D_{in}^{(t)}*D_{out}^{(t)}))$ , wherein  $K_t$  denotes the temporal kernel size, while  $D_{in}^{(t)}$  and  $D_{out}^{(t)}$  represent the input and output dimensions of TCL, respectively. The time complexity of the GCL is characterized as  $O(TN^2K_s*D_{in}^{(s)}*D_{out}^{(s)})$ , with  $K_s$  being the spatial kernel size. Moreover,  $D_{in}^{(s)}$  and  $D_{out}^{(s)}$  represent the input and output dimensions of GCL. By treating  $K_t$ ,  $K_s$ ,  $D_{in/out}^{(t)}$ , and  $D_{in/out}^{(s)}$  as constants, we deduce that the time complexity of the TSRL module amounts to  $O(NT+TN^2)$ .
- CD: The computation amount associated with the CD module equals O(NTM\*D), where M corresponds to the number of negative samples and D is the dimension of traffic representation. We treat D as constants and obtain the time complexity as O(NTM).
- CO-SSD: The computation amount of the CO-SSD module comprises two principal segments: a 1-D convolution network and a two-layer MLP. Their respective complexities are  $O(NT*D_{in}^{(c)}*D_{out}^{(c)})$  and  $O(N*D_{in}^{(m)}*D_{out}^{(m)})$ . Analogous to earlier observations, considering  $D_{in/out}^{(c)}$  and  $D_{in/out}^{(m)}$  as constants enables us to infer that the time complexity of the CO-SSD module stands at O(NT+N).

Tab. VIII depicts the practical model efficiency considering both training and inference phases. Compared to the best baseline AGCRN, our proposed STEVE reduces the training and inference time by 66.3% and 73.1% on average. This

efficiency improvement enhances the practical applicability of our proposed model.

#### V. RELATED WORK

Spatio-Temporal Traffic Forecasting. ST data-based traffic forecasting has received increasing attention due to its pivotal role in Intelligent Transportation Systems [2]. Early contributions [28], [29] emerged from the time series community and predominantly utilized the ARIMA family to model ST traffic data. However, these methods usually rely on stationary assumptions, leading to limited representation power for ST traffic data. Recent advancements have seen the application of diverse deep learning techniques, which are free from stationary assumptions, to capture complex traffic dependencies. For instance, methods like recurrent neural networks [30]-[32] and temporal convolutional networks [6]-[8] have been employed to capture temporal dependencies. As for spatial dependencies, convolutional neural networks [17], [18] have been employed for grid-based ST data; graph neural networks [1], [2], [33], [34] and attention mechanism [10], [35]-[37] have been explored to introduce road network information. Recently, some studies explored the OOD generalization of ST models, focusing on invariant relation learning [24], external factors modeling [4], or temporal OOD scenarios [38]. Differently, this paper develops a principled approach to enhance ST forecasting with better robustness in both spatial and temporal OOD scenarios via self-supervised deconfounding.

Self-Supervised Learning aims to distill valuable information from input data to enhance the quality of representations [39]. The fundamental paradigm involves initially augmenting input data and subsequently employing self-supervised tasks to serve as pseudo labels for the purpose of representation learning [13], [40], [41]. These tasks are usually infused with domain knowledge to encourage representations to exhibit specific characteristics. This approach has achieved remarkable success within various data such as text [42], image [43], and audio data [44]. Motivated by these works, we devise customized self-supervised learning tasks tailored to infuse spatio-temporal context information into traffic data representations. This remains relatively unexplored for OOD ST forecasting.

**Out-Of-Distribution (OOD) Generalization** is aimed at tackling scenarios where the distributions in the test phase are different from those in the training phase [45], [46]. This issue, although prevalent, poses a significant challenge across multiple domains, including computer vision [47], [48], natural language processing [49], and time series analysis [4], [20], [50]. Within the realm of spatio-temporal traffic forecasting, the OOD phenomenon naturally arises due to different ST contexts from which training and test data are generated. Despite the existence of various techniques tailored for OOD generalization in other domains [22], [51]–[53], this issue has received limited attention in the context of ST traffic forecasting.

Causal Inference serves as a tool to identify causal relations among variables, thereby facilitating stable and robust learning and inference [9], [54]. This approach has demonstrated significant accomplishments across domains such as image data analysis [55], [56], text data processing [57], and user behavior data modeling [58], [59]. A recent work [4] makes an attempt to apply causal inference theory to ST data. However, this approach requires external ST context data, which might not always be accessible. In contrast, our proposed method offers a comprehensive solution that leverages the ST context data generated simultaneously with traffic observations to facilitate the learning of causal representations.

#### VI. CONCLUSION AND FUTURE WORK

This paper investigated the problem of spatio-temporal (ST) forecasting for out-of-distribution (OOD) urban traffic data. We first formalized this widespread problem and proposed a theoretical scheme named disentangled contextual adjustment (DCA) from a causal perspective. It leveraged *do*-intervention to deconfound non-causal relations by modeling the effect of invariant and variant ST contexts separately. To implement DCA, we developed a deep learning framework called STEVE. It learned context-oriented disentangled traffic representations for OOD ST forecasting by incorporating context information into ST dependency modeling in a self-supervised fashion. Extensive experiments on four benchmark traffic datasets demonstrated the robustness of our STEVE in OOD traffic scenarios. Our proposed model also achieves better scalability and efficiency compared to the state-of-the-art methods.

In real-world urban traffic data, hidden confounders often hinder causal inference from observed data. In future work, we aim to explore the integration of instrumental variables into our STEVE to further enhance the estimation of causal relations, particularly in the absence of contextual data.

#### PROOF FOR THEORETICAL SCHEME

### A. Proof of Rule 2

Since Rule 1 is straight-forwarding, we only provide the derivation of Rule 2 as follows.

*Proof.* Conditional probability is a measure of the probability of an event occurring, given that another event (by assumption, presumption, assertion or evidence) has already occurred <sup>1</sup>. Rule 2 can be derived by the properties of the conditional probability.

Since both equations in Rule 2 share the same forms, we take the first equation, i.e.,  $P(C_I = \mathcal{C}_{I_k}) = \frac{P(C = \mathcal{C}_{I_k})}{P(C_I)}$ , as an example for the proof. Considering event  $A = \{C = \mathcal{C}_{I_k}\}, I_k \in [I_1, I_K]$ , and event  $B = \{C = \mathcal{C}_I\} = \{C = \mathcal{C}_{I_1}, C = \mathcal{C}_{I_2}, \ldots, C = \mathcal{C}_{I_K}\}$ , we have  $A \cap B = \{C = \mathcal{C}_{I_k}\} = A$ . The right term of the above-mentioned equation can be expressed as:

$$\frac{P(C=\mathcal{C}_{I_k})}{P(C_I)} = \frac{P(C=\mathcal{C}_{I_k})}{P(C=\mathcal{C}_I)} = \frac{P(A\cap B)}{P(B)} = P(A|B). \quad (21)$$

Here, we use Rule 1 in the first step and the definition of conditional probability in the third step. Then, we substitute

<sup>1</sup>https://en.wikipedia.org/wiki/Sample\_space

event A and B with their definitions and use Rule 1 again to obtain:

$$P(A|B) = P(C = C_{I_k}|C = C_I) = P(C = C_{I_k}|C_I).$$
 (22)

Since  $C_I \subset C$ , we have

$$P(C = \mathcal{C}_{I_k}|C_I) = P(C_I = \mathcal{C}_{I_k}). \tag{23}$$

We derive the left term from the right term of the first equation of Rule 2 via Eq. (21)-(23), thereby proving their equivalence. Also, the second equation can be proved through the same procedure.

# B. Proof of Theorem 1

*Proof.* Since each ST context can be categorized as invariant or variant according to its major membership, we divide the context set of Eq. (3) into two groups and calculate them separately:

$$P_{\Theta}(Y|do(X)) = \sum_{I_{k}=I_{1}}^{I_{K}} P_{\Theta}(Y|X, C = C_{I_{k}}) P(C = C_{I_{k}}) + \sum_{V_{k}=V_{1}}^{V_{K}} P_{\Theta}(Y|X, C = C_{V_{k}}) P(C = C_{V_{k}}).$$
(24)

For each group, we introduce  $P(C_I)$  and  $P(C_V)$  as follows:

$$P_{\Theta}(Y|do(X)) = \sum_{I_{k}=I_{1}}^{I_{K}} P_{\Theta}(Y|X, C = C_{I_{k}}) \frac{P(C = C_{I_{k}})}{P(C_{I})} P(C_{I}) + \sum_{V_{k}=V_{1}}^{V_{K}} P_{\Theta}(Y|X, C = C_{V_{k}}) \frac{P(C = C_{V_{k}})}{P(C_{V})} P(C_{V}).$$
(25)

Since  $P(C_I)$  and  $P(C_V)$  are constant w.r.t. indexing variables  $I_k$  and  $V_k$ , we move them to the outside of the summation:

$$P_{\Theta}(Y|do(X)) = P(C_I) \sum_{I_k=I_1}^{I_K} P_{\Theta}(Y|X, C = C_{I_k}) \frac{P(C = C_{I_k})}{P(C_I)} + P(C_V) \sum_{V_k=V_1}^{V_K} P_{\Theta}(Y|X, C = C_{V_k}) \frac{P(C = C_{V_k})}{P(C_V)}.$$
(26)

Then, we apply Rule 2 and have

$$P_{\Theta}(Y|do(X)) = P(C_I) \sum_{I_k=I_1}^{I_K} P_{\Theta}(Y|X, C = C_{I_k}) P(C_I = C_{I_k}) + P(C_V) \sum_{V_k=V_1}^{V_K} P_{\Theta}(Y|X, C = C_{V_k}) P(C_V = C_{V_k}).$$
(27)

Next, we apply the law of total probability and obtain

$$P_{\Theta}(Y|do(X)) = P(C_I)P_{\Theta}(Y|X, C = \mathcal{C}_I) + P(C_V)P_{\Theta}(Y|X, C = \mathcal{C}_V).$$
(28)

We then come to our proposed disentangled contextual adjustment (DCA) in Eq. (4) by applying Rule 1 to Eq. (28). This means that we can successfully estimate  $P_{\Theta}(Y|do(X))$  by our proposed DCA in Theorem 1.

#### REFERENCES

- L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," vol. 33, 2020, pp. 17804–17815.
- [2] J. Ji, J. Wang, Z. Jiang, J. Jiang, and H. Zhang, "STDEN: Towards physics-guided neural networks for traffic flow prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4048–4056.
- [3] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," in 2016 IEEE 16th international conference on data mining (ICDM). IEEE, 2016, pp. 499–508.
- [4] Y. Zhao, P. Deng, J. Liu, X. Jia, and M. Wang, "Spatial-temporal neural structural causal models for bike flow prediction," in *Proceedings of the* AAAI Conference on Artificial Intelligence, 2023.
- [5] W. Jiang, "Bike sharing usage prediction with deep learning: a survey," Neural Computing and Applications, vol. 34, no. 18, pp. 15369–15385, 2022.
- [6] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *The 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.
- [7] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 2018, pp. 3634–3640.
- [8] J. Wang, J. Ji, Z. Jiang, and L. Sun, "Traffic flow prediction based on spatiotemporal potential energy fields," *IEEE Transactions on Knowledge* and Data Engineering, pp. 1–14, 2022.
- [9] M. Glymour, J. Pearl, and N. P. Jewell, Causal inference in statistics: A primer. John Wiley & Sons, 2016.
- [10] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, 2022.
- [11] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference:* foundations and learning algorithms. The MIT Press, 2017.
- [12] Y. Hagmayer, S. A. Sloman, D. A. Lagnado, and M. R. Waldmann, "Causal reasoning through intervention," *Causal learning: Psychology, philosophy, and computation*, pp. 86–100, 2007.
- [13] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, and Y. Zheng, "Spatio-temporal self-supervised learning for traffic flow prediction," in Proceedings of the AAAI Conference on Artificial Intelligence, 2023.
- [14] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "CLUB: A contrastive log-ratio upper bound of mutual information," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, 13-18 July 2020, Virtual Event, 2020, pp. 1779–1788.
- [15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," stat, vol. 1050, p. 1, 2014.
- [16] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference* on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, F. R. Bach and D. M. Blei, Eds., 2015, pp. 1180–1189.
- [17] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the Thirty-First* AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, 2017, pp. 1655–1661.
- [18] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 February 1, 2019. AAAI Press, 2019, pp. 5668–5675.
- [19] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "St-norm: Spatial and temporal normalization for multi-variate time series forecasting," in *Proceedings of the 27th ACM SIGKDD conference on knowledge* discovery & data mining, 2021, pp. 269–278.
- [20] Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, and C. Wang, "Adarnn: Adaptive learning and forecasting of time series," in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 402–411.

- [21] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," in *International Conference on Learning Representations*, 2022
- [22] Y. Chen, Y. Zhang, Y. Bian et al., "Learning causally invariant representations for out-of-distribution generalization on graphs," Advances in Neural Information Processing Systems, vol. 35, pp. 22131–22148, 2022.
- [23] P. Deng, Y. Zhao, J. Liu, X. Jia, and M. Wang, "Spatio-temporal neural structural causal models for bike flow prediction," arXiv preprint arXiv:2301.07843, 2023.
- [24] Z. Zhou, Q. Huang, K. Yang, K. Wang, X. Wang, Y. Zhang, Y. Liang, and Y. Wang, "Maintaining the status quo: Capturing invariant relations for ood spatiotemporal learning," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, p. 3603–3614.
- [25] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proceedings of the IEEE/CVF conference on computer* vision and pattern recognition, 2019, pp. 1871–1880.
- [26] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied* mathematics, vol. 20, pp. 53–65, 1987.
- [27] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." Journal of machine learning research, vol. 9, no. 11, 2008.
- [28] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal arima model with limited input data," *European Transport Research Review*, vol. 7, no. 3, pp. 1–9, 2015.
- [29] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert systems with applications*, vol. 36, no. 3, pp. 6164– 6173, 2009.
- [30] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.
- [31] X. Tang, H. Yao, Y. Sun, C. Aggarwal, P. Mitra, and S. Wang, "Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values," in AAAI, vol. 34, no. 04, 2020, pp. 5956–5963.
- [32] Z. Fang, L. Pan, L. Chen, Y. Du, and Y. Gao, "Mdtp: A multi-source deep traffic prediction framework over spatio-temporal trajectory data," *Proceedings of the VLDB Endowment*, vol. 14, no. 8, pp. 1289–1297, 2021.
- [33] 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," in AAAI, vol. 35, no. 17, 2021, pp. 15008–15015.
- [34] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, and C. S. Jensen, "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2733–2746, 2022.
- [35] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 914–921.
- [36] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference* on artificial intelligence, vol. 34, no. 01, 2020, pp. 1234–1241.
- [37] Y. Cui, K. Zheng, D. Cui, J. Xie, L. Deng, F. Huang, and X. Zhou, "Metro: a generic graph neural network framework for multivariate time series forecasting," *Proceedings of the VLDB Endowment*, vol. 15, no. 2, pp. 224–236, 2021.
- [38] Y. Xia, Y. Liang, H. Wen, X. Liu, K. Wang, Z. Zhou, and R. Zimmer-mann, "Deciphering spatio-temporal graph forecasting: A causal lens and treatment," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [39] J. Ji, J. Wang, J. Wu, B. Han, J. Zhang, and Y. Zheng, "Precision cityshield against hazardous chemicals threats via location mining and self-supervised learning," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3072–3080.
- [40] H. Ren, J. Wang, and W. X. Zhao, "Generative adversarial networks enhanced pre-training for insufficient electronic health records modeling," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3810–3818.

- [41] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Proceedings of the AAAI conference* on artificial intelligence, vol. 31, no. 1, 2017.
- [42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, 2019, pp. 4171–4186.
- [43] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International* conference on machine learning. PMLR, 2020, pp. 1597–1607.
- [44] A. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.
- [45] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, "Towards out-of-distribution generalization: A survey," arXiv preprint arXiv:2108.13624, 2021.
- [46] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International conference on machine learning*. PMLR, 2013, pp. 10–18.
- [47] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu, "Generalizing to unseen domains: A survey on domain generalization," *IEEE Transactions on Knowledge and Data Engineering*, 2022
- [48] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4396–4415, 2023.
- [49] A. Ramponi and B. Plank, "Neural unsupervised domain adaptation in NLP—A survey," in *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Dec. 2020, pp. 6838–6855.
- [50] H. Yao, C. Choi, B. Cao, Y. Lee, P. W. W. Koh, and C. Finn, "Wild-time: A benchmark of in-the-wild distribution shift over time," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10309–10324, 2022.
- [51] C. Liu, X. Sun, J. Wang, H. Tang, T. Li, T. Qin, W. Chen, and T.-Y. Liu, "Learning causal semantic representation for out-of-distribution prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 6155–6170, 2021.
- [52] T. Wang, Z. Yue, J. Huang, Q. Sun, and H. Zhang, "Self-supervised learning disentangled group representation as feature," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18225–18240, 2021.
- [53] C. Yang, Q. Wu, Q. Wen, Z. Zhou, L. Sun, and J. Yan, "Towards out-of-distribution sequential event prediction: A causal treatment," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.
- [54] J. Pearl, "Models, reasoning, and inference." Cambridge University Press, 2000.
- [55] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, "Causal intervention for weakly-supervised semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 655–666, 2020.
- [56] X. Deng and Z. Zhang, "Comprehensive knowledge distillation with causal intervention," Advances in Neural Information Processing Systems, vol. 34, pp. 22 158–22 170, 2021.
- [57] Y. Niu, K. Tang, H. Zhang, Z. Lu, X.-S. Hua, and J.-R. Wen, "Counterfactual vqa: A cause-effect look at language bias," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12700–12710.
- [58] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, "Disentangling user interest and conformity for recommendation with causal embedding," in *Proceedings of the Web Conference* 2021, 2021, pp. 2980–2991.
- [59] X. He, Y. Zhang, F. Feng, C. Song, L. Yi, G. Ling, and Y. Zhang, "Addressing confounding feature issue for causal recommendation," ACM Transactions on Information Systems, vol. 41, no. 3, pp. 1–23, 2023.