STONE: A Spatio-temporal OOD Learning Framework Kills Both Spatial and Temporal Shifts

Binwu Wang
University of Science and Technology of China
Hefei, China
wbw1995@mail.ustc.edu.cn

Jiaming Ma*
University of Science and Technology of China
Hefei, China
jiamingma@mail.ustc.edu.cn

Pengkun Wang
Suzhou Institute for Advanced Research, University of Science and Technology of China
Suzhou, China
pengkun@mail.ustc.edu.cn

Xu Wang
Suzhou Institute for Advanced Research, University of Science and Technology of China
Suzhou, China
wx309@mail.ustc.edu.cn

Yudong Zhang
University of Science and Technology of China
Hefei, China
zyd2020@mail.ustc.edu.cn

Zhengyang Zhou
Suzhou Institute for Advanced Research, University of Science and Technology of China
Suzhou, China
zzy0929@mail.ustc.edu.cn

Yang Wang*
University of Science and Technology of China
Hefei, China
angyan@ustc.edu.cn

ABSTRACT

Traffic prediction is a crucial task in the Intelligent Transportation System (ITS), receiving significant attention from both industry and academia. Numerous spatio-temporal graph convolutional networks have emerged for traffic prediction and achieved remarkable success. However, these models have limitations in terms of generalization and scalability when dealing with Out-of-Distribution (OOD) graph data with both structural and temporal shifts. To tackle the challenges of spatio-temporal shift, we propose a framework called STONE by learning invariable node dependencies, which achieve stable performance in variable environments. STONE initially employs gated-transformers to extract spatial and temporal semantic graphs. These two kinds of graphs represent spatial and temporal dependencies, respectively. Then we design three techniques to address spatio-temporal shifts. Firstly, we introduce a Fréchet embedding method that is insensitive to structural shifts, and this embedding space can integrate loose position dependencies of nodes within the graph. Secondly, we propose a graph intervention mechanism to generate multiple variant environments by perturbing two kinds of semantic graphs without any data augmentations, and STONE can explore invariant node representation from environments. Finally, we further introduce an explore-to-extrapolate risk objective to enhance the variety of generated environments. We conduct experiments on multiple traffic datasets, and the results demonstrate that our proposed model exhibits competitive performance in terms of generalization and scalability.

CCS CONCEPTS

• **Computing methodologies** → *Temporal reasoning*; • **Information systems** → *Data mining*.

KEYWORDS

Spatio-temporal data mining, Out-of-distribution generalization, Traffic prediction, Causal graph learning

ACM Reference Format:
Binwu Wang, Jiaming Ma*, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang*. 2024. STONE: A Spatio-temporal OOD Learning Framework Kills Both Spatial and Temporal Shifts. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671680

1 INTRODUCTION

With the increasing prevalence of GPS-enabled mobile devices and sensors, a significant amount of spatio-temporal data is being gathered in various fields such as urban transportation and atmospheric conditions. This data has become a valuable asset, fueling progress in the realm of urban computing [7, 22, 38, 41, 42, 54, 57, 63]. Among the many applications, predicting traffic flow is a standout task in urban computing, providing dependable future road insights and enhancing traffic management systems [20, 26, 27, 56, 65].

Yang Wang and Jiaming Ma are corresponding authors.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671680

Current prevailing traffic flow prediction method processes data into graph-structured data incorporating relationship induction, namely spatio-temporal graphs (STG), and then spatio-temporal graph convolutional networks are employed as engines to learn spatio-temporal features and make prediction [25, 39, 43, 59]. Given a graph $\mathbf{G}$ and training data $\mathbf{X}$ denoted as the training environment $e : \{\mathbf{X}, \mathbf{G}\}$, the goal of STG prediction is to learn a function $\mathcal{F}$ which can predict target label y given associated input x:

$$\min_{\mathcal{F}} \mathbb{E}_{(x,y)\sim \mathrm{P}(x,y|e)} [\mathcal{L}(\mathcal{F}(\mathbf{x}), y)|e]. \tag{1}$$

While existing models have been highly successful, they heavily depend on the IID assumption, which states that testing and training data are independently sourced from the same environment. Unfortunately, many cutting-edge models like D^2 STGNN [31] and PDFormer [9] are coupled with the training STG. However, this assumption does not always hold in real-world settings where spatial and temporal features of STG may change over time, leading to varying testing environments and posing challenges related to Out-of-Distribution (OOD) scenarios. A few recent studies [8, 48, 64] have started exploring methods for OOD spatio-temporal learning. Yet, these methods primarily focus on temporal shifts and overlook significant spatial changes. In this paper, we first comprehensively define spatio-temporal shift from two concepts: **temporal shift** and **spatial shift**, as shown in Figure. 1.

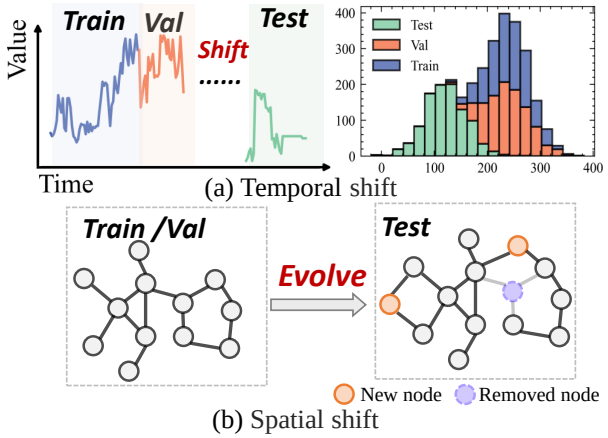

Figure 1: Spatio-temporal shift is interpreted into the temporal and spatial shifts. Temporal shift refers to the changes in the distribution feature (such as mean and variance) of traffic data over time. Spatial shift refers to the evolution of the graph structure, typically involving changes in the graph's size.

(1) Temporal shift refers to the change in the temporal data distribution (e.g., mean and variance), meaning that $\mathrm{P}(\mathbf{X}_{\mathrm{te}})$ is not equal to $\mathrm{P}(\mathbf{X}_{\mathrm{tr}})$. For example, Figure 1(a) shows the traffic flow of a node (sensor) in the PeMS system over a long period and reveals its shifted flow distribution.

(2) Spatial shift refers to the evolution of underlying graph structures, as depicted in Figure 1(b). In a modern transportation system, the road network tends to gradually expand over time. New nodes may emerge due to increasing sensors, while existing nodes may disappear due to engineering renovations or equipment failures.

Further, we formally formalize the spatio-temporal OOD challenge as, $e_{\mathrm{tr}} : \{\mathbf{X}_{\mathrm{tr}}, \mathbf{G}_{\mathrm{tr}}\} \neq e_{\mathrm{te}} : \{\mathbf{X}_{\mathrm{te}}, \mathbf{G}_{\mathrm{te}}\}$ with $\mathrm{P}(\mathbf{X}_{\mathrm{te}}) \neq \mathrm{P}(\mathbf{X}_{\mathrm{tr}})$ and $\mathbf{G}_{\mathrm{te}} \neq \mathbf{G}_{\mathrm{tr}}$. When dealing with the OOD challenge, we argue that existing models have two limitations: *inflexible scalability* and *unreliable generalization*. One major reason behind these limitations is that they use GCN to learn node representations on pre-defined training graphs, and this learned knowledge is tied to these specific graphs, which cannot accurately represent unseen graphs. In particular, when dependencies between nodes change dynamically due to spatial-temporal shifts, such as the addition or removal of nodes, the representations aggregated through the original dependency paths fail to respond to these shifts, substantially impeding their generalization ability in various environments. Moreover, GCN cannot effectively generalize the learned knowledge to emerging nodes that are unknown to the model during the training phase. thus, the scalability of models in variable environments also poses a significant challenge. Traffic management personnel might be particularly interested in the traffic conditions of these nodes to devise updated scheduling plans.

Causal learning has garnered considerable attention in dealing with OOD problems within the image and NLP fields [30, 62]. Current approaches in these domains primarily focus on extracting stable knowledge that can consistently perform well across diverse data distributions. However, spatio-temporal OOD tasks present two major challenges: (1) How to leverage GCN to learn reliable representations that are resilient to spatio-temporal shifts. (2) How to devise intervention mechanisms to explicitly enhance environment environmental modeling for robust generalization.

In this paper, we propose a novel causal graph learning framework for spatio-temporal OOD learning. The key idea is to extract invariant spatio-temporal dependencies among nodes that can consistently represent the relationship of nodes across various STG distributions, enabling a reliable aggregation path that empowers GCN to learn generalizable representations irrespective of specific graphs. To generate a range of distributions, we model spatio-temporal shift by perturbing learned node dependencies instead of manipulating the spatio-temporal graph data.

Specifically, we propose **S**patio-**T**emporal **O**OD Graph Learning **N**etworks with Fréchet **E**mbedding (STONE) for spatio-temporal OOD learning. STONE comprises two main components: a semantic graph learning component and a graph intervention mechanism. The first component of STONE utilizes a transformer with a gate to effectively capture spatio-temporal heterogeneity and generate two kinds of semantic graphs that represent dependencies among nodes in the dimensions of temporal and spatial, respectively. To extract stable dependencies, we first employ a Fréchet embedding method to encode the topology information of the graphs. The embedding space serves as a loose mapping of the global position of nodes within the graphs and exhibits flexibility for spatial shifts. Secondly, we design a spatio-temporal graph intervention mechanism that involves perturbing two generated semantic graphs. This perturbation process effectively simulates spatio-temporal shifts, thereby creating variable environments. By extracting invariant spatio-temporal dependencies from these environments, the model

is guided by these dependencies as aggregated paths to learn a robust representation. Finally, we introduce an explore-to-extrapolate risk term to enhance the variety of generated environments, enabling the model to explore and extrapolate beyond the observed data, thereby improving its ability to handle unseen distributions. Experiment results on various OOD traffic datasets demonstrate that our model achieves competitive performance in generalization performance and scalability. The main contribution of this paper can be summarized as follows:

- To the best of our knowledge, we are the first to comprehensively investigate spatio-temporal OOD learning, considering both temporal shift and structural shift.
- We propose a novel framework called STONE, which aims to learn invariant node dependencies between nodes. This allows the model to maintain consistent prediction performance in variable environments.
- This framework incorporates several innovative components, including a novel Fréchet embedding, a graph intervention mechanism, and an intervention loss term. These components are designed to enhance the generalization and scalability of the model.
- Experimental results on multiple real-world traffic datasets demonstrate that our model achieves competitive generalization and scalability across various OOD scenarios.

2 RELATED WORK

2.0.1 Spatiotemporal graph prediction. Traffic prediction plays a vital role in the intelligent transportation system domain [10–12, 18, 40]. Currently, the prevalent approach transforms traffic data into spatio-temporal graphs and employs cutting-edge spatio-temporal graph convolutional networks to handle intricate spatio-temporal dynamics [14, 17, 34–37, 49, 55]. Notably, D^2STGNN [31] combines diffusion graph convolutional networks with RNNs to capture temporal patterns effectively. STGCN [53] utilizes TCN to efficiently model time dependencies. HGC-RNN [52] leverages optimization techniques based on hypergraph convolution, while STSGCN [33] employs local graph convolution to address large-scale graph scenarios. Nevertheless, existing models may exhibit suboptimal performance when dealing with OOD challenges. This limitation stems from their fundamental assumption that the test and training environments are drawn from the same distribution.

2.0.2 OOD graph learning. Several methods have been proposed in the field of graph representation learning to enhance generalization performance for OOD problems [28, 46]. For instance, GAUG [58] improves downstream training and inference processes by modifying the input graph using an edge prediction module. DisenGCN [24] focuses on learning representations that disentangle distinct and informative factors within the graph data, assigning these factors to different parts of the factorized vector representations. OOD-GNN [16] introduces a nonlinear graph representation decorrelation approach utilizing random Fourier features to eliminate statistical dependence between causal and noncausal graph representations generated by the graph encoder.

2.0.3 OOD learning in the time domain. There are some OOD learning models in the time domain to address shifts of time series data. For example, AdaRNN [5] clusters historical time sequences into different classes and dynamically matches input data to these classes to identify contextual information. Other invariance learning models for sequential data are commonly used to learn disentangled seasonal-trend representations [44] or environment-specific representations [50]. DIVERSIFY [23] attempts to exploit subdomains within a whole dataset to counteract issues induced by non-stationary generalized representation learning. However, these models fail to model spatial dependencies.

2.0.4 Spatio-temporal OOD learning with temporal shift. Influenced by the advancements in OOD graph learning within the recommendation domain, researchers have recently moved their attention towards exploring the problem of OOD with temporal shifts. For example, CauSTG [64] presents a causal framework that is capable of transferring local and global spatio-temporal invariant relations to out-of-distribution scenarios. CaST [48] utilizes a causal model (SCM) to interpret the data generation process of spatio-temporal graphs. It employs back-door adjustment to separate the invariant components from the temporal environment. STEVE [6] encodes traffic data into two disentangled representations and utilizes spatio-temporal environments as self-supervised signals to incorporate contextual information into these representations. This enhances the generalization ability of the learned context-oriented representations, thereby improving OOD generalization. However, the current research primarily focuses on investigating the effects of temporal drift and fails to consider the evolution of the graph structure.

3 PROBLEM PRELIMINARIES

3.0.1 Spatio-temporal graph. We use a graph structure to represent spatio-temporal data denoted as $\mathbf{G} = (V, E, \mathbf{A})$, where V means the node set with N nodes and E means the set of edges, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix. We use $x_t \in \mathbb{R}^{N\times d}$ to represent the historical graph data of N nodes at t-th time step, where d means the number of features.

3.0.2 Spatio-temporal graph prediction. Given a training environment e comprising two parts: the graph $\mathbf{G}$ and the training data $\mathbf{X}$, this task aims to learn a prediction function $\mathcal{F}$ that takes the observed data of the past T time steps as input, $\mathbf{x} \in \mathbb{R}^{T\times N\times d} = \{x_1, \cdots, x_T\}$ sampled from $\mathbf{X}$, to predict the future data in T_P time steps. The parameters of $\mathcal{F}$ are optimized by calculating the loss between the predicted value and the ground truth y as follows:

$$\min_{\mathcal{F}} \mathbb{E}_{(\mathbf{x},y)\sim \mathrm{P}(x,y|e)} [\mathcal{L}(\mathcal{F}(\mathbf{x}), y)]. \tag{2}$$

3.0.3 Spatio-temporal OOD learning. The goal of spatio-temporal OOD learning is to find a function $\mathcal{F}$ that effectively makes predictions for given input graph data from any of support environments $\mathcal{E}$.

$$\min_{\mathcal{F}} \max_{e^*\in\mathcal{E}} \mathbb{E}_{(\mathbf{x},y)\sim \mathrm{P}(\mathbf{x},y|e^*)} [\mathcal{L}(\mathcal{F}(\mathbf{x}), y)]. \tag{3}$$

Definition: Self-attention mechanism SA $(\cdot)$**.** Self-attention mechanism can learn the global correlation between different positions in the input sequence, which is widely applied in the field of time series analysis [61]. Given an input $\mathbf{H} \in \mathbb{R}^{N\times d_h}$, the self-attention

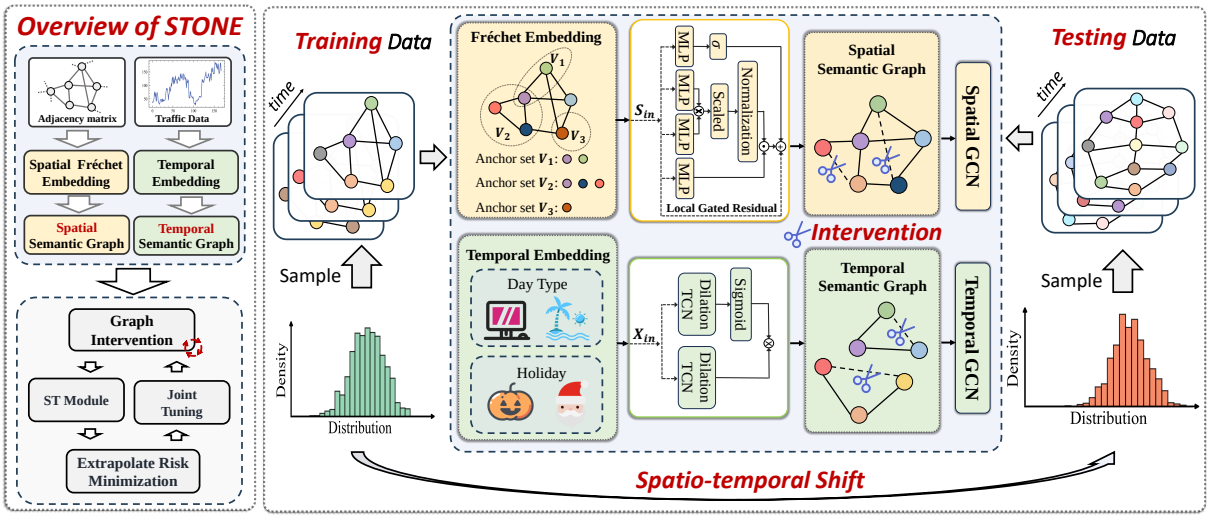

Figure 2: The details of the proposed model. Our framework first models spatio-temporal heterogeneity and extracts two semantic graphs, and then we perturb semantic graphs to create variable environments, enabling the model to learn invariant dependencies from these environments.

mechanism SA $(\cdot)$ computes the attention coefficient by the dot product operation:

$$\mathrm{SA}(\mathbf{H}) = \mathrm{Softmax}\left[\frac{(\mathbf{H}W_q + b_q)(\mathbf{H}W_k + b_k)^\top}{\sqrt{d_x}}\right]. \quad (4)$$

where $W_q, W_k \in \mathbb{R}^{d_h \times d_x}$, b_q, and b_k are learnable parameters. We can obtain an attention matrix with size $N \times N$.

4 METHOD

In this section, we present the details of STONE, as illustrated in Figure 2 and Algorithm 25. We provide an overview of the framework and subsequently explain each component of the model individually.

4.1 Overview of STONE

STONE consists of three main modules that contribute to its functionality: a semantic graph learning module for acquiring semantic graphs from input data and prior knowledge, a spatio-temporal graph convolution module for consolidating information across semantic graphs to make predictions, and a graph intervention mechanism that aims to create diverse spatio-temporal environments where the first two modules can extract invariable knowledge.

Spatio-temporal graph learning module. The input data $\mathbf{x} \in \mathbb{R}^{T \times N \times d}$ and the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ of the graph are passed through embedding layers, which integrates static prior information into the model. Subsequently, gated transformers are employed to effectively model spatio-temporal heterogeneity. This process generates two types of semantic graphs: a spatial semantic graph denoted as D_s, and a temporal semantic graph denoted as D_t. D_t captures the similarity of time-varying features between nodes, providing insights into their temporal dependencies. On the other hand, D_s encodes graph topology information, describing the affinity between nodes based on their positions within the graph.

We use spatiotemporal graph convolutional networks to aggregate temporal and spatial information from these two semantic graphs separately, and the generated node representations are denoted as X_o and S_o. Finally, a gated decoder is employed to decode X_o and S_o for making predictions.

Graph intervention mechanism. We design a graph interference mechanism for spatio-temporal OOD learning, and this mechanism perturbs two generated semantic graphs, D_s and D_t by randomly masking edges within them. In fact, This process simulates spatial and temporal shifts, resulting in the creation of diverse intervention environments. By extracting invariant spatio-temporal dependencies from these environments, spatio-temporal graph convolutional networks can generate generalizable node representations. Further, we introduce an explore-to-extrapolate risk term to enhance the variety of intervention environments which can prompt the model to explore and extrapolate beyond the observed data.

4.2 Spatial Semantic Graph Learning

4.2.1 Node embedding. Node embedding technique has been proven to enhance prediction performance by incorporating graph topology information [60]. However, traditional node embedding methods struggle with spatial shifts. For instance, Random-walk based methods, such as node2vec, sample local substructure, and this is sensitive to shifts.

To solve this problem, we develop a novel Fréchet embedding. This method can preserve graph local and global context information by encoding the distance affinity between the nodes and the selected anchor points, which can well response to the dynamic nature of the graph. Specifically, given the adjacency matrix $\mathbf{A}$ of a graph, this method can return a low-dimensional vector $S_{in} \in \mathbb{R}^{N \times d_e}$.

Definition 1: Fréchet Embedding. For a metric space (V, d_V), we define an function $f : V \to \ell_p^{d_e}$ as *Fréchet embedding*, if each of the

coordinates f_i is proportional to the distance with anchor sets $\mathcal{V}$, that is,

$$f_i(u) = \alpha_i \cdot d_V(u, \mathcal{V}_i) = \alpha_i \cdot \min_{v \in V_i} d_V(u, v), \quad \forall u \in V, \forall i = 1, \ldots, d_e. \tag{5}$$

where $\mathcal{V}_i$ means i-th anchor set with d_e anchors, and a_i is a numeric factor. V is the node set, and a metric function $d_V(\cdot)$ on the set V is called an ℓ_p metric if there exists a natural number d_e and an embedding of (V, d) into the space $\ell_p^{d_e}$. For $p = 2$, ℓ_2 would be an Euclidean metric. A pseudometric ℓ_p is defined similarly.

Proposition 1: Spatio-temporal graph Fréchet embedding. To define spatio-temporal graph Fréchet embedding, we select K-order Manhattan distance [29] as the metric d_V with nodes set V of G to form the metric space (V, d_V). Each dimension of the Fréchet embedding represents the minimum of the metric between nodes concerning a fixed subset $\mathcal{V}$ of the set of nodes V, and we call such a fixed subset the anchor sets. We obtain $\lceil logN \rceil$ anchor sets for each resampling, and a total of R rounds of resampling are performed to obtain $R \times \lceil logN \rceil$ anchor sets $\{\mathcal{V}\}_1^{R \times \lceil logN \rceil}$. In each round $i \in \{1, \ldots, R\}$, we sample $N * 2^{-j}$ nodes into anchor set $\mathcal{V}_{ij}$[1] for $j \in \{1, 2, ..., \lceil logN \rceil\}$, following a specific rule: for j-th anchor set $\mathcal{V}_{ij}$ in i-th resampling round, any node u in V is selected as anchor node in $\mathcal{V}_{ij}$ with probability 2^{-j}. Given the metric space (V, d_V) and $\{\mathcal{V}\}_1^{R \times \lceil logN \rceil}$, we define spatio-temporal graph Fréchet embedding as $f : V \to \ell_p^{R \times \lceil logN \rceil}$, where $f_{ij}(u) = \alpha_{ij} d_V^{(k)}(u, \mathcal{V}_{ij})$, where α_{ij} is a learnable parameter with $\sum_{i=1}^{R} \sum_{j=1}^{\lceil logN \rceil} \alpha_{ij}^p = 1$ for given metric space ℓ_p.

Property. The Fréchet embedding can preserve the structural information of the original metric space by the relative positions between the perceptual nodes and the selected anchor set. This means that closely connected nodes within the graph also have close embeddings, thus the embedding space has good generalization and scalability. On the one hand, spatial shifts do not lead to significant deviations in this embedding space, hence the node dependencies are elastic; on the other hand, new nodes can easily obtain good initial embedding in this space by calculating their distances to the anchors, thereby enhancing the model's scalability. We carefully illustrate this through ablation experiments in 5.4 and visual studies in 5.5.

The Fréchet embedding, in reality, is not isometric; it undergoes slight changes in distances while maintaining the graph structure to a certain extent. These alterations in the distances reflect the deviation between the embedding space and the original space. Subsequently, we will demonstrate that our embedding is characterized by a low distortion upper limit of $O(logN)$.

Definition 2: Distortion. Given a metric space (V, d_V) and embedding metric space (Y, d_Y), an injective mapping $f : X \to Y$ is called a D-embedding, where $D \geq 1$ is a real number if there exists a constant $r > 0, a \geq 1$:

$$r \cdot d_V(u, v) \leq d_Y(f(u), f(v)) \leq Dr \cdot d_V(u, v), \quad \forall u, v \in V. \tag{6}$$

The infimum of the numbers D such that f is a D-embedding is called the distortion of f.

[1]The value $\lceil \log N \rceil$ can guarantee that the expected number of anchor nodes per set is greater than 1 and achieves less distortion.

Theorem 1: Bourgain theorem. This tells us that the mapping function f in spatio-temporal graph Fréchet embedding constructed by the random sample algorithm satisfies:

$$\frac{1}{O(logN)} d_{mh}^{(k)}(u, v) \leq \mathbb{E}_f \parallel fu - fv \parallel_{lp} \leq d_V^{(k)}(u, v), \forall u, v \in V. \tag{7}$$

Thus, the distortion of the embedding is $O(logN)$.

4.2.2 Spatial gated transformer for graph leanring. Given the output from the spatial Fréchet embedding layer $S_{in} \in \mathbb{R}^{N \times d_e}$ where $d_e = \lceil logN \rceil \times R$, we further propose a spatial gated transformer to extract deep embedding. Then, we use the self-attention mechanism to generate a spatial semantic graph.

Specifically, we use two MLP layers to map S_{in} into another dimensional feature space as follows:

$$S^{(1)} = ReLU\left(S_{in} W_1^{(0)} + b_1^{(0)}\right) W_2^{(0)} + b_2^{(0)}. \tag{8}$$

where $W_1^{(0)}, b_1^{(0)}, W_2^{(0)}$, and $b_2^{(0)}$ are learnable parameters. Then a spatial gated transformer uses the self-attention function SA $(\cdot)$ to calculate the similarity between nodes: $\alpha_S^{(l)} = \text{SA}\left(S^{(l)}\right) \in \mathbb{R}^{N \times N}$, where $S^{(l)} \in \mathbb{R}^{N \times d_s^{(l)}}$ means the input of l-th layer. The output vectors are then fused to obtain spatial representations with a gate:

$$\begin{aligned} S^{(l+1)} &= \sigma_S^{(l)} \odot \text{ReLU}\left[\alpha^{(l)}\left(S^{(l)} W_v^{(l)} + b_v^{(l)}\right)\right] + \left(1 - \sigma_S^{(l)}\right) \odot S^{(l)}, \\ \sigma_S^{(l)} &= \text{Sigmoid}\left[S^{(l)} W_\sigma^{(l)} + b_\sigma^{(l)}\right] \in \mathbb{R}^{N \times d_s^{(l+1)}}. \end{aligned} \tag{9}$$

where $W_v^{(l)}, b_v^{(l)}, W_\sigma^{(l)}$, and $W_\sigma^{(l)}$ are learnable parameters. $S^{(l+1)} \in \mathbb{R}^{N \times d_e^{(l+1)}}$ is the output spatial representation. $\sigma_S^{(l)}$ is a gate to filter out redundant information. After two spatial gated transformer layers, the output embedding vector is denoted as S_s. Then we use the attention mechanism to generate spatial semantic graphs D_s:

$$D_s = \text{SA}(S_s) \in \mathbb{R}^{N \times N}. \tag{10}$$

where this spatial semantic graph D_s encodes relative position relationships of nodes.

4.3 Temporal semantic graph learning

4.3.1 Temporal position embedding. This module encodes temporal prior information such as the date type and holiday type of the input sequence into $\mathbf{x} \in \mathbb{R}^{T \times N \times d}$, this information can help the model better analyze temporal trends. The output of this layer is denoted as $X_T^0 \in \mathbb{R}^{T \times N \times d_t}$.

4.3.2 Temporal gated convolution for graph learning. We use the TCN architecture to extract temporal trends from spatio-temporal graph data of each node. TCN is configured with L_t causal convolution networks with different receptive fields to model long- and short-term dependencies. We also introduce a gate to improve performance. Specifically, given the input $X^{(l)} \in \mathbb{R}^{N \times T_l \times d_t}$ in the l-the layer, where T_l means the time-step length of input time series, the forward process is shown as follows:

$$\begin{aligned} X_T^{(l+1)} &= \sigma_t^l \odot \left(\theta_{k_t} *_t X_T^{(l)}\right) \in \mathbb{R}^{N \times T_{l+1} \times d_t}, \\ \sigma_T^l &= Sigmoid\left[\theta_{k_d} *_t X_T^{(l)}\right] \in \mathbb{R}^{N \times T_{l+1} \times d_t}. \end{aligned} \tag{11}$$

where θ_{k_t} and θ_{k_d} are learnable parameters, k_t and k_d are kernel sizes of causal convolution networks. σ_T^l is a gated and the residual connection technique is used for smooth learning. Finally, we splice the output of each layer to and merge information of all time-steps:

$$\begin{aligned} X_t &= ReLU\left[X_T^{(0:L_t)} \times_2 W_t^1\right] \times_2 W_t^2 \in \mathbb{R}^{N\times 1\times C_t}, \\ X_T^{(0:L_t)} &= X_T^{(0)} \oplus X_T^{(1)} \oplus \ldots \oplus X_T^{(L_t)} \in \mathbb{R}^{N\times\left(\sum_{l=0}^{L_t} T_l\right)\times C_t}. \end{aligned} \tag{12}$$

where $W_t^1 \in \mathbb{R}^{\left(\sum_{l=0}^{L_t} T_l\right)\times d_w^t}$ and $W_t^2 \in \mathbb{R}^{d_w^t\times 1}$ are learnable parameters, $\times_2$ means tensor multiplication in the second dimension and $\oplus$ means the concatenation of tensors. Finally, we also use the attention mechanism to calculate the similarity between nodes:

$$D_t = \mathrm{SA}\left(X_t\right) \in \mathbb{R}^{N\times N}. \tag{13}$$

where the similarity matrix $D_t \in \mathbb{R}^{N\times N}$ means the correlation of the flow distribution between nodes.

4.4 Spatio-temporal Graph Convolutional Network

GCN has been shown to be effective in processing graph data in multiple tasks [2**?** –4]. Given learned temporal feature vector X_t with temporal semantic matrix D_t and spatial feature vector S_i with the spatial semantic matrix D_s, we use diffusion graph convolution with Z diffusion steps as spatio-temporal graph convolutional networks to aggregate the information of two dimensions separately. We exchange the two matrices to fuse spatio-temporal information:

$$\begin{aligned} X_o &= \sum_{i=1}^{Z} \theta_s *_s \left(I_s + D_s\right)^i X_t \in \mathbb{R}^{N\times d_o}, \\ S_o &= \sum_{i=1}^{Z} \theta_t *_s \left(I_t + D_t\right)^i S_s \in \mathbb{R}^{N\times d_o}. \end{aligned} \tag{14}$$

where θ_s and θ_t are learnable kernel parameters. I_s and I_t denote the identity matrices of D_t and D_s, respectively.

4.5 Spatio-temporal OOD learning

The traditional spatio-temporal learning process only allows the model to adapt to the specific training environment, which limits its ability to handle out-of-distribution data with spatio-temporal shifts. To overcome this limitation, it is necessary to expose the model to diverse training environments, enabling it to learn invariant node representations. However, directly generating variable out-of-distribution spatio-temporal data is computationally complex. To tackle this challenge, we propose a novel graph intervention mechanism that perturbs generated semantic graphs, thereby simulating training data from variable environments. Additionally, we introduce a loss term that encourages the model to explore and extrapolate beyond the observed data, enhancing its capability to handle unseen scenarios.

4.5.1 Noise disturbance. We randomly add noise $\gamma \sim \pi(0,1)$, which is drawn from the standard normal distribution, into the output vector S_s from the Fréchet embedding layer. This process changes the spatial dependencies of nodes and simulates spatial shifts, which can help the model explore a more robust embedding space.

4.5.2 Spatio-temporal graph intervention mechanism. To illustrate the graph intervention mechanism, let's consider the spatial semantic D_s as an example. We create an intervention matrix $M_s \in [0,1]^{N\times N}$, where each row u follows a binomial distribution $\mathcal{B}\left(1, p\left(u\right)\right)$. The probability $p\left(u\right)$ is calculated using the softmax function with learnable parameters π_u. Thus, its i-th row and j-th column $M_S\left[i,j\right]$ is a binary to indicate whether $D_s\left[i,j\right]$ is masked. The setting that each row in M_S is sampled from the same learnable binomial distribution is to improve computational efficiency. Then, By performing the dot product between M_S and D_s, we can generate a new adjacency matrix $\hat{D_s}$, which would be used in the graph convolution operation in Equ.14. This matrix can be viewed as a changed spatial dependence caused by spatial shifts.

In an ideal scenario, the training environment would encompass all possible data distributions. However, achieving such an exhaustive coverage is not impractical. To move towards this objective, our aim is to enrich the training environment, enabling the model to learn from a broader spectrum of data distributions. This enhancement facilitates better adaptation to unforeseen circumstances. Specifically, we create K_M intervention matrices denoted as $\mathbb{M} = \left\{M_s^1, \cdots, M_s^{K_M}\right\}$, where K_M is a hyper-parameter. Similar to the spatial semantic graph D_s, we also perform a comparable intervention strategy on the temporal semantic graph D_t.

4.6 Decoder and optimization loss

We use a gate unit with MLP layers as a decoder to predict future graph data:

$$\begin{aligned} \sigma_{out} &= \mathrm{Sigmoid}\left[S_o W_{out}^s + X_o W_{out}^i\right] \in \mathbb{R}^{K_M\times N\times d_{out}}, \\ \hat{Y} &= \sigma_{out} \odot \left(X_o W_{out}^o + b_{out}^o\right) \in \mathbb{R}^{K_M\times N\times T_p}. \end{aligned} \tag{15}$$

where W_{out}^s, W_{out}^i, W_{out}^o, and b_{out}^o are learnable parameters. T_p means the prediction window length. The prediction loss between predicted values $\hat{Y}$ and ground-truth values $y \in \mathbb{R}^{N\times T_p}$ can be computed as follows:

$$\mathcal{L}\left(y|\mathbb{M},\Theta\right) = \frac{1}{K_M}\sum_{m=1}^{K_M} \mathcal{L}_\Theta\left(\hat{Y}\left[m\right], y\right) = \frac{1}{K_M}\sum_{m=1}^{K_M} \left\|\hat{Y}\left[m\right] - y\right\|_{l_1}. \tag{16}$$

where $\hat{Y}\left[m\right] \in \mathbb{R}^{N\times T_p}$ means the m-th row of $\hat{Y}$ and Θ is the parameter set of prediction function $\mathcal{F}$.

To enhance the extraction of consistent representations from various simulated environments, we propose the Invariant Risk Minimization (IRM) objective [1]. Furthermore, the presence of diverse environments can enhance the model's capacity to generalize to unfamiliar distributions. Therefore, we incorporate an Explore-to-Extrapolate risk [45] to increase the variance of the intervention matrix, enabling thorough exploration of environments and promoting robust learning of the models. Hence, the optimization loss function we employ is:

$$\begin{aligned} &\min_{\Theta} \mathrm{Var}\left(\mathcal{L}\left(y \mid \mathbb{M}^*, \Theta\right)\right) + \beta\mathcal{L}\left(y \mid \mathbb{M}^*, \Theta\right), \\ s.t. \quad &\mathbb{M}^* = \left\{M_*^1, \ldots, M_*^{K_M}\right\} = \underset{m\in\{1,\cdots,K_M\}}{\mathrm{argmax}} \mathrm{Var}\left\{\mathcal{L}\left(y \mid M^m, \Theta\right)\right\}. \end{aligned} \tag{17}$$

where Var(·) means the loss variance, β is a trade-off to balance two loss terms.

5 EXPERIMENT

In this section, we conduct a comprehensive evaluation of the generalization performance (Section 5.2) and scalability performance (Section 5.3) of STONE[2]. We then perform ablation experiments in Section 5.4 to verify the validity of each component. Additionally, we visualize the Fréchet embedding to study its properties in Section 5.5 and analyze the learned semantic graph in Section 5.6.

5.1 Datasets and setting

5.1.1 Original dataset. In our experiments, we utilize two datasets, namely the SD and GBA datasets, to evaluate the effectiveness of STONE. These datasets are subsets of the LargeST dataset [21], which records the traffic flow data from thousands of sensors spanning the period from 2017 to 2021 in the Caltrans Performance Measurement System (PeMS). SD and GBA datasets collected traffic information from 716 and 2352 sensors, respectively, where their spatio-temporal graphs are constructed based on the travel distance between sensors.

5.1.2 Data processing. We use two processing methods to process SD and GBA to imitate both spatial and temporal shifts for testing.
Temporal shift. We choose the data from 1/2019-8/2019 with 21010 timestamps for training, 9/2019-10/2019 with 7003 timestamps as validation data, and 11/2020-12/2020 with 7022 timestamps as testing data. The ratio of these three subsets is about 6:2:2.
Spatio-temporal shift. With the temporal shift, we select a certain number of nodes for training, which amounts to 550 in the SD dataset and 1809 in the GBA dataset. For validation sets, We select 10% of the number of training nodes from the remaining nodes to add to training graphs and then mask 10% of them. For testing sets, we consider three different ratios of new nodes compared to training graphs: 10%, 15%, and 20%. Additionally, we also randomly mask 10% of nodes in testing sets to simulate node disappearance. Two spatio-temporal shift datasets with r% new nodes are denoted as STSD-r and STGBA-r, where $r \in \{10, 15, 20\}$. The details of these datasets are presented in Table 1.

Table 1: Details of STSD and STGBA datasets.

Dataset		STSD	STGBA
Training	Time span	1/2019-8/2019	1/2019-8/2019
	Nodes	550	1809
Val	Time span	9/2019-10/2019	9/2019-10/2019
	Removed Nodes	55	180
	New Nodes	55	180
Test	Time span	11/2020-12/2020	11/2020-12/2020
	Removed Nodes	55	180
	New Nodes	55/82/110	180/270/360

[2]The code is available at https://github.com/PoorOtterBob/STONE-KDD-2024, where also provides the pseudocode for STONE.

5.1.3 Model setting. We set the batch size to 64 and use the Adam optimizer [15] with a learning rate of $1e^{-3}$. The trade-off parameter, β, of the loss function is set to 1. We stacked two gated transformer layers to generate the semantic graphs. In the intervention mechanism, we create two intervention matrices, i.e., $K_m = 2$. In the Fréchet embedding, we perform 10 sampling rounds in the STSD dataset and 30 sampling rounds in the STGBA dataset. We evaluate the performance of the models using three widely used metrics: MAE, RMSE, and MAPE at 3, 6, and 9 horizon. Models are directly tested on testing sets after training without further fine-tuning.

5.1.4 Baselines. We compare our proposed method, STONE, with SOTA traffic prediction models and spatio-temporal OOD learning methods. However, it is important to note that some SOTA models cannot be run under the spatio-temporal shift setting, as their core component parameters are coupled to the scale of the graph, such as D^{2}STGNN [31] and PDFormer [9]. The traffic prediction models include Historical Average (HL), GWNet [47], STGCN [53], and STNN [51]. For spatio-temporal OOD learning methods, we compare against CauSTG [64] and CaST [48].

5.2 Generalization performance of models

The prediction performance of all nodes on the STSD and STGBA datasets with varying rates of new nodes is presented in Table 2.

In the realm of various OOD spatio-temporal datasets, GWNet achieved relatively lower errors, likely attributed to its usage of diffusion graph convolutional networks. This enables bidirectional modeling of spatio-temporal dependencies, thereby improving its capacity to adapt to shifts in space and time. Conversely, STGCN exhibited higher prediction errors by relying solely on information aggregation according to a predetermined graph structure, making it vulnerable to spatial changes. Consequently, its predictive performance becomes increasingly limited with the increase in new nodes. In contrast, STNN outperforms STGCN in prediction accuracy by incorporating attentional mechanisms to capture general spatio-temporal correlations independent of a specific training graph. Nevertheless, its prediction accuracy is somewhat compromised due to the assumption of independently and identically distributed data. On the other hand, CaST is designed to address temporal shifts specifically but encounters challenges in effectively accommodating spatial shifts. These models demonstrate superior performance in STGBA datasets with larger graphs, offering more comprehensive spatio-temporal insights.

Our model achieved competitive prediction performance in various OOD scenarios, with a maximum improvement of up to 18.13% in terms of MAPE. This improvement can be attributed to our approach of creating a range of training environments by perturbing the learned semantic graphs. By training on these distributions, STONE can effectively learn generalizable knowledge, which keeps consistent performance across OOD scenarios.

5.3 Scalability performance of models

We evaluated the scalability of our model by reporting the prediction performance of new nodes in the testing datasets. The experiment results are shown in Table 3. We observe that GCN-based models can perform neighborhood aggregation mechanisms to generate representations for new nodes. However, models like STNN and

Table 2: Generalization performance of each model in OOD traffic datasets with spatio-temporal shifts. The best results are marked in bold and the second best results are underlined.

STSD dataset with ratio of new nodes: (10%/15%/20%)								
Model		HL	STGCN [53]	GWNet [47]	STNN [51]	CaST [48]	CauSTG [64]	Ours
3 horizon	MAE	29.66/29.78/29.69	26.62/24.88/23.52	18.86/21.58/**19.12**	39.82/40.23/35.58	24.23/24.05/23.89	26.42/25.31/26.17	**18.16**/**19.36**/19.32
	RMSE	44.55/44.66/44.52	36.49/34.55/34.84	**29.40**/**30.25**/34.27	58.02/58.66/54.59	38.42/38.06/37.75	40.01/40.17/39.89	29.67/31.42/**31.40**
	MAPE	21.43/21.45/21.41	48.51/40.54/30.76	18.90/18.62/18.44	39.53/39.27/20.76	20.73/20.79/30.76	23.04/21.38/22.16	**15.50**/**16.69**/**16.87**
6 horizon	MAE	52.05/52.31/52.19	34.35/33.32/32.05	27.15/27.01/27.62	39.78/40.20/35.82	35.96/35.71/35.49	40.01/41.76/40.93	**26.32**/**26.23**/**26.24**
	RMSE	75.30/75.51/75.34	**40.38**/42.82/45.97	42.89/41.70/41.35	57.77/58.43/54.91	55.47/55.00/54.63	60.34/66.21/66.31	41.76/**40.79**/**40.91**
	MAPE	39.43/39.41/39.36	55.77/49.54/43.31	30.88/30.36/30.12	40.40/40.13/44.21	36.35/36.47/36.39	41.46/41.82/42.17	**22.08**/**26.16**/**26.48**
12 horizon	MAE	94.13/94.61/94.44	45.28/47.07/46.71	**39.31**/39.14/40.32	43.02/43.42/41.51	61.63/61.21/60.86	64.13/66.00/65.34	41.05/**36.17**/**36.19**
	RMSE	128.11/128.59/128.36	64.04/64.67/65.64	58.09/57.63/59.60	62.31/62.89/63.39	90.39/89.74/89.19	89.42/90.16/91.02	**55.26**/**53.79**/**54.10**
	MAPE	82.08/81.94/81.85	56.49/57.87/68.57	45.47/44.89/44.62	43.36/43.10/48.88	60.40/60.72/60.52	65.37/65.31/65.04	**35.55**/**38.57**/**38.94**
STGBA dataset with ratio of new nodes: (10%/15%/20%)								
Model		HL	STGCN [53]	GWNet [47]	STNN [51]	CaST [48]	CauSTG [64]	Ours
3 horizon	MAE	27.09/26.94/26.97	19.36/25.19/35.65	19.23/**17.56**/18.53	37.48/37.33/37.55	26.86/26.78/26.82	30.14/30.41/31.13	**17.67**/18.73/**18.83**
	RMSE	40.37/40.16/40.15	29.13/34.78/36.27	30.01/29.34/32.62	54.31/54.18/54.43	37.05/36.95/36.99	41.03/42.86/43.02	**27.84**/**29.98**/**30.13**
	MAPE	18.90/18.86/18.83	15.82/26.85/30.33	13.71/14.77/**12.86**	31.78/31.94/31.96	34.83/35.20/37.94	36.13/37.40/36.93	**12.82**/**12.84**/12.91
6 horizon	MAE	47.04/46.80/46.85	25.68/33.73/35.86	28.10/25.24/26.71	37.07/36.93/37.15	36.89/36.78/36.85	40.35/41.14/41.66	**25.35**/**25.16**/**25.37**
	RMSE	67.46/67.14/67.15	43.77/46.36/48.49	42.78/**37.81**/40.28	53.85/53.71/53.97	51.37/51.23/51.29	55.39/55.49/55.43	**37.80**/38.55/**38.88**
	MAPE	34.53/34.45/34.41	21.38/34.40/39.09	20.88/20.41/19.09	31.55/31.69/31.73	43.37/43.80/43.50	46.15/46.18/46.24	**18.17**/**18.21**/**18.39**
12 horizon	MAE	84.85/84.45/84.50	**34.50**/48.59/50.90	39.91/38.94/39.24	41.16/41.03/41.23	58.87/58.67/58.79	63.15/64.28/64.05	36.35/**36.04**/**36.51**
	RMSE	114.83/114.34/114.36	59.81/66.16/68.18	57.91/56.40/57.33	59.67/59.55/59.77	81.25/80.99/81.12	88.31/89.35/89.02	**53.20**/**52.80**/**53.55**
	MAPE	70.53/70.73/70.25	33.18/46.94/51.19	30.53/32.80/28.83	34.16/34.31/34.36	65.61/66.31/65.79	70.14/70.64/70.64	**28.38**/**28.47**/**28.74**

STGCN exhibited larger prediction errors, potentially because their parameters are coupled with the road network structure, resulting in insufficient information to generate accurate representations for emerging nodes. GWNet achieved better prediction performance by utilizing diffusion graph convolutional networks, which aggregate bidirectional information to provide more comprehensive insights.

In contrast, our proposed model achieved optimal scalability. This is because our model learns a robust spatio-temporal semantic graph structure, enabling accurate predictions by perceiving the semantic neighborhood information of unseen nodes.

5.4 Ablation experiment

To evaluate the effectiveness of each contribution in the model, we conducted ablation experiments on the STSD-10% dataset. We created three variations: (1) W/O IL, where we removed the intervention loss term. (2) W/O Emb, where we removed the Fréchet embedding. (3) W/O Noi, where we no longer added random noise to the node embeddings after the Fréchet embedding layer.

The results of the 3-horizon prediction are shown in Table 4. W/O IL achieved worse errors because the intervention loss term helps improve the diversity of the variable environment, thereby enhancing the model's generalization performance. W/O Emb had higher errors because the GCN, which is sensitive to structural shifts, failed to generate accurate representations for the evolved graph. On the other hand, the Fréchet embedding space loosely preserves the graph's structural information, making semantic neighbors more robust. This property also improves the scalability performance for new nodes by perceiving their semantic neighbors. In summary, each variant performed inferior to the proposed model, demonstrating the effectiveness of each component.

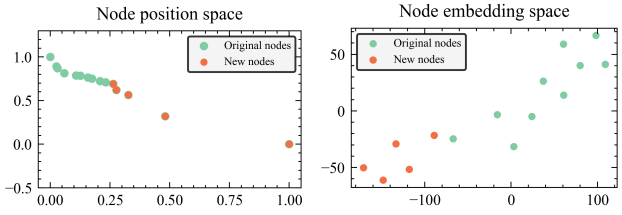

Figure 3: Node position and Fréchet embedding visualization.

5.5 Fréchet embedding study

We selected several nodes from the STSD-10% dataset and extracted their embedding vectors from S_{in}, which are the output of the Fréchet embedding module. Then, we applied the t-SNE technique to reduce the dimensions of these vectors. Figure 3 visualizes the positions and embedding vectors of these nodes. It is evident that the Fréchet embedding effectively preserves the structural information of the graph. Nodes that are close in the graph also have close embeddings in space, indicating the preservation of proximity relationships. Additionally, the Fréchet embedding space is elastic to the shift of the graph structure, as the addition or removal of nodes does not cause significant changes in the embedding space.

5.6 Semantic graph visualization study

We extracted optimized spatial and temporal semantic graphs from STONE trained in STSD-10%. Figure 4 displays the edge weights of 10 nodes across these two graphs along with the predefined graph based on geographic location. In the predefined graph, the

Table 3: Scalability performance of each model in OOD traffic datasets with spatio-temporal shifts. The best results are marked in bold and the second best results are underlined.

ST-SD dataset with ratio of new nodes: (10%/15%/20%)								
Model		HL	STGCN [53]	GWNet [47]	STNN [51]	CaST [48]	CauSTG [64]	Ours
3 Horzion	MAE	29.64/30.57/29.85	36.19/34.46/29.59	19.58/19.58/19.58	42.56/44.52/36.37	23.37/22.24/21.75	42.56/44.52/39.37	**17.22**/**18.92**/**18.51**
	RMSE	43.26/44.51/43.69	48.46/42.91/41.29	30.11/**30.05**/29.76	61.74/65.06/55.62	36.11/34.11/33.27	61.74/65.06/59.62	**26.85**/30.43/**29.83**
	MAPE	21.48/21.56/21.32	48.74/39.05/29.69	19.02/19.35/**18.38**	37.23/36.00/25.38	19.40/20.30/20.19	37.23/36.00/25.38	**14.93**/**18.67**/18.60
6 Horzion	MAE	51.56/53.70/52.63	51.81/45.34/43.07	28.48/28.33/28.55	42.72/44.68/36.63	35.05/33.36/32.69	42.72/44.68/36.63	**24.91**/**26.31**/**25.71**
	RMSE	73.30/75.84/74.71	71.92/62.80/60.62	42.82/41.85/41.59	61.88/65.06/55.95	53.09/50.34/49.28	61.88/65.06/55.95	**38.60**/**40.71**/**40.01**
	MAPE	40.63/39.79/39.39	61.74/53.30/42.74	30.42/31.71/29.10	38.11/36.70/**25.19**	32.96/34.98/34.88	38.11/36.70/25.19	**22.03**/**30.77**/30.43
12 Horzion	MAE	93.11/97.12/95.46	76.75/68.71/66.13	41.98/41.40/42.12	45.01/47.31/42.13	59.80/57.15/56.06	45.01/47.31/42.13	**39.53**/**37.78**/**36.74**
	RMSE	125.86/130.04/128.33	109.81/95.18/92.64	63.23/60.96/61.35	65.25/68.45/64.07	86.98/83.12/81.46	65.21/68.42/64.03	**59.37**/**56.51**/**55.37**
	MAPE	85.42/83.31/82.39	74.37/75.54/60.52	46.56/44.25/43.58	40.51/43.05/42.82	52.78/87.73/57.30	41.33/43.02/44.11	**34.71**/**36.95**/**34.27**
ST-GBA dataset with ratio of new nodes: (10%/15%/20%)								
Model		HL	STGCN [53]	GWNet [47]	STNN [51]	CaST [48]	CauSTG [64]	Ours
3 Horzion	MAE	26.67/25.65/26.16	26.14/32.97/35.14	21.64/**18.00**/21.39	36.98/36.41/37.70	25.88/25.60/26.17	30.15/27.21/29.87	**17.64**/18.25/**19.59**
	RMSE	39.20/37.91/38.44	35.91/46.41/46.11	32.79/**28.59**/32.98	52.98/52.94/54.37	35.19/34.98/35.72	40.04/39.15/40.61	**27.03**/29.73/**30.16**
	MAPE	18.26/18.13/18.19	21.99/30.60/35.59	14.62/14.87/16.48	29.24/31.54/31.58	30.26/34.56/33.17	34.13/35.16/36.03	**12.81**/**13.02**/**13.15**
6 Horzion	MAE	46.54/44.86/45.62	37.28/47.88/52.03	33.01/25.89/33.46	36.51/35.94/37.23	35.62/35.19/35.98	38.75/39.01/38.40	**25.73**/**24.87**/**25.46**
	RMSE	65.93/63.96/64.76	50.74/66.20/68.08	48.41/39.40/51.00	52.40/52.32/53.78	49.12/48.78/49.74	55.46/56.14/55.03	**37.55**/**38.82**/**39.54**
	MAPE	33.55/33.29/33.31	32.07/42.58/51.97	22.99/20.55/27.13	29.01/31.27/31.34	37.31/42.68/41.14	40.14/47.43/43.27	**18.61**/**18.69**/**19.08**
12 Horzion	MAE	84.24/81.34/82.46	57.31/74.33/80.60	49.10/39.92/53.09	39.89/39.46/40.86	56.85/55.97/57.40	61.24/64.39/65.15	**38.24**/**36.50**/**37.60**
	RMSE	112.84/109.71/110.98	77.88/101.28/105.41	69.42/58.67/79.15	67.16/57.30/58.88	78.24/77.21/78.94	84.32/86.42/85.22	**56.56**/**54.26**/**55.81**
	MAPE	68.55/70.37/67.85	47.28/63.89/74.58	35.51/32.68/43.06	31.04/33.31/33.67	54.58/63.70/61.18	59.10/68.91/64.32	**29.86**/**29.84**/**30.23**

Table 4: Ablation experiment on STSD-10% dataset.

Model	Generalization (All nodes)			Scalability (New nodes)		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Ours	**18.17**	**29.67**	**15.51**	**17.23**	**26.85**	**14.93**
W/O Emb	21.91	32.24	26.79	21.64	31.84	25.30
W/O IL	19.66	31.29	17.83	20.49	32.23	20.49
W/O Noi	19.36	31.43	16.69	18.92	30.43	18.67

correlations between nodes are scattered, which means that a spatio-temporal shift can disrupt the aggregation of neighborhood information and propagate through the entire graph via message passing mechanisms. In learned semantic graphs, nodes primarily establish connections with a small number of crucial nodes. As a result, the addition or removal of nodes has minimal impact. Even if some crucial nodes are removed, the model can still aggregate information from the remaining nodes in both temporal and spatial dimensions, leading to the generation of accurate representations.

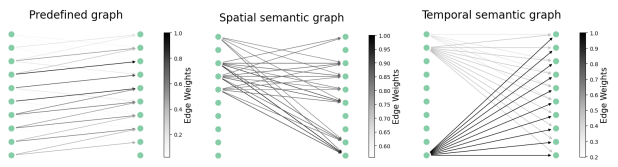

Figure 4: Edge connectivity of 10 nodes in three graphs.

6 CONCLUSION

In this paper, we introduce a new framework STONE for spatio-temporal OOD learning. STONE integrates a semantic graph learning module to capture spatial heterogeneity and generate semantic graphs in both temporal and spatial dimensions. Then we propose a graph intervention mechanism to perturb the generated semantic graph to create diverse training environments. With an Explore-to-Extrapolate loss term, STONE can extract stable spatio-temporal aggregation information paths, thereby generating invariant spatio-temporal representations, which can generalize well to unknown environments. We conduct extensive experiments to evaluate the effectiveness of STONE. The results demonstrate that STONE achieves competitive performance in terms of both generalization and scalability.

ACKNOWLEDGMENTS

This paper is supported by the National Natural Science Foundation of China (No.62072427, No.12227901), the Project of Stable Support for Youth Teamin Basic Research Field, CAS (No.YSBR-005), and Academic Leaders Cultivation Program, USTC.

REFERENCES

[1] Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893* (2019).
[2] Chen, H., Xu, Y., Huang, F., Deng, Z., Huang, W., Wang, S., He, P., and Li, Z. Label-aware graph convolutional networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020), p. 1977–1980.
[3] Du, W., Chen, L., Wang, H., Shan, Z., Zhou, Z., Li, W., and Wang, Y. Deciphering urban traffic impacts on air quality by deep learning and emission inventory. *journal of environmental sciences 124* (2023), 745–757.
[4] Du, W., Yang, X., Wu, D., Ma, F., Zhang, B., Bao, C., Huo, Y., Jiang, J., Chen, X.,

and Wang, Y. Fusing 2d and 3d molecular graphs as unambiguous molecular descriptors for conformational and chiral stereoisomers. *Briefings in Bioinformatics 24*, 1 (2023), bbac560.

[5] Du, Y., Wang, J., Feng, W., Pan, S., Qin, T., Xu, R., and Wang, C. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management* (2021), pp. 402–411.

[6] Hu, J., Liang, Y., Fan, Z., Chen, H., Zheng, Y., and Zimmermann, R. Graph neural processes for spatio-temporal extrapolation. *arXiv preprint arXiv:2305.18719* (2023).

[7] Huang, Q., Shen, L., Zhang, R., Cheng, J., Ding, S., Zhou, Z., and Wang, Y. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2024), vol. 38, pp. 12608–12616.

[8] Ji, J., Zhang, W., Wang, J., He, Y., and Huang, C. Self-supervised deconfounding against spatio-temporal shifts: Theory and modeling. *arXiv preprint arXiv:2311.12472* (2023).

[9] Jiang, J., Han, C., Zhao, W. X., and Wang, J. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. *arXiv preprint arXiv:2301.07945* (2023).

[10] Jin, G., Li, F., Zhang, J., Wang, M., and Huang, J. Automated dilated spatio-temporal synchronous graph modeling for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* (2022).

[11] Jin, G., Liang, Y., Fang, Y., Shao, Z., Huang, J., Zhang, J., and Zheng, Y. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[12] Jin, G., Xi, Z., Sha, H., Feng, Y., and Huang, J. Deep multi-view graph-based network for citywide ride-hailing demand prediction. *Neurocomputing 510* (2022), 79–94.

[13] Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., and Tang, J. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (2020), pp. 66–74.

[14] Jin, Y., Chen, K., and Yang, Q. Transferable graph structure learning for graph-based traffic forecasting across cities. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023), pp. 1032–1043.

[15] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Li, H., Wang, X., Zhang, Z., and Zhu, W. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[17] Li, H., Zhao, Y., Mao, Z., Qin, Y., Xiao, Z., Feng, J., Gu, Y., Ju, W., Luo, X., and Zhang, M. A survey on graph neural networks in intelligent transportation systems. *arXiv preprint arXiv:2401.00713* (2024).

[18] Li, Z., Xia, L., Tang, J., Xu, Y., Shi, L., Xia, L., Yin, D., and Huang, C. Urbangpt: Spatio-temporal large language models. *arXiv preprint arXiv:2403.00813* (2024).

[19] Liang, Y., Ouyang, K., Wang, Y., Liu, Y., Zhang, J., Zheng, Y., and Rosenblum, D. S. Revisiting convolutional neural networks for citywide crowd flow analytics. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I* (2021), Springer, pp. 578–594.

[20] Liu, C., Yang, S., Xu, Q., Li, Z., Long, C., Li, Z., and Zhao, R. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134* (2024).

[21] Liu, X., Xia, Y., Liang, Y., Hu, J., Wang, Y., Bai, L., Huang, C., Liu, Z., Hooi, B., and Zimmermann, R. Largest: A benchmark dataset for large-scale traffic forecasting. *arXiv preprint arXiv:2306.08259* (2023).

[22] Liu, Z., Miao, H., Zhao, Y., Liu, C., Zheng, K., and Li, H. Lighttr: A lightweight framework for federated trajectory recovery. *arXiv preprint arXiv:2405.03409* (2024).

[23] Lu, W., Wang, J., Sun, X., Chen, Y., Ji, X., Yang, Q., and Xie, X. Diversify: A general framework for time series out-of-distribution detection and generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[24] Ma, J., Cui, P., Kuang, K., Wang, X., and Zhu, W. Disentangled graph convolutional networks. In *International conference on machine learning* (2019), PMLR, pp. 4212–4221.

[25] Miao, H., Fei, Y., Wang, S., Wang, F., and Wen, D. Deep learning based origin-destination prediction via contextual information fusion. *Multimedia Tools and Applications* (2022), 1–17.

[26] Miao, H., Shen, J., Cao, J., Xia, J., and Wang, S. Mba-stnet: Bayes-enhanced discriminative multi-task learning for flow prediction. *TKDE* (2022).

[27] Miao, H., Zhao, Y., Guo, C., Yang, B., Kai, Z., Huang, F., Xie, J., and Jensen, C. S. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *ICDE* (2024).

[28] Park, H., Lee, S., Kim, S., Park, J., Jeong, J., Kim, K.-M., Ha, J.-W., and Kim, H. J. Metropolis-hastings data augmentation for graph neural networks. *Advances in Neural Information Processing Systems 34* (2021), 19010–19020.

[29] Peiravi, A., and Kheibari, H. T. A fast algorithm for connectivity graph approximation using modified manhattan distance in dynamic networks. *Applied mathematics and computation 201*, 1-2 (2008), 319–332.

[30] Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE 109*, 5 (2021), 612–634.

[31] Shao, Z., Zhang, Z., Wei, W., Wang, F., Xu, Y., Cao, X., and Jensen, C. S. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *arXiv preprint arXiv:2206.09112* (2022).

[32] Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine 30*, 3 (2013), 83–98.

[33] Song, C., Lin, Y., Guo, S., and Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence* (2020), vol. 34, pp. 914–921.

[34] Wang, B., Wang, P., Zhang, Y., Wang, X., Zhou, Z., Bai, L., and Wang, Y. Towards dynamic spatial-temporal graph learning: A decoupled perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2024), vol. 38, pp. 9089–9097.

[35] Wang, B., Wang, P., Zhang, Y., Wang, X., Zhou, Z., and Wang, Y. Condition-guided urban traffic co-prediction with multiple sparse surveillance data. *IEEE Transactions on Vehicular Technology* (2024).

[36] Wang, B., Zhang, Y., Wang, P., Wang, X., Bai, L., and Wang, Y. A knowledge-driven memory system for traffic flow prediction. In *International Conference on Database Systems for Advanced Applications* (2023), Springer, pp. 192–207.

[37] Wang, B., Zhang, Y., Wang, X., Wang, P., Zhou, Z., Bai, L., and Wang, Y. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023), pp. 2223–2232.

[38] Wang, J., Jiang, J., Jiang, W., Han, C., and Zhao, W. X. Towards efficient and comprehensive urban spatial-temporal prediction: A unified library and performance benchmark. *arXiv preprint arXiv:2304.14343* (2023).

[39] Wang, L., Guo, D., Wu, H., Li, K., and Yu, W. Tc-gcn: Triple cross-attention and graph convolutional network for traffic forecasting. *Information Fusion* (2024), 102229.

[40] Wang, S., Cao, J., Chen, H., Peng, H., and Huang, Z. Seqst-gan: Seq2seq generative adversarial nets for multi-step urban crowd flow prediction. *ACM Transactions on Spatial Algorithms and Systems (TSAS) 6*, 4 (2020), 1–24.

[41] Wang, S., Miao, H., Chen, H., and Huang, Z. Multi-task adversarial spatial-temporal networks for crowd flow prediction. In *CIKM* (2020), pp. 1555–1564.

[42] Wang, S., Miao, H., Li, J., and Cao, J. Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks. *TITS 23*, 5 (2021), 4695–4705.

[43] Wang, X., Wang, P., Wang, B., Zhang, Y., Zhou, Z., Bai, L., and Wang, Y. Latent gaussian processes based graph learning for urban traffic prediction. *IEEE Transactions on Vehicular Technology* (2023).

[44] Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. *arXiv preprint arXiv:2202.01575* (2022).

[45] Wu, Q., Zhang, H., Yan, J., and Wipf, D. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466* (2022).

[46] Wu, Y.-X., Wang, X., Zhang, A., He, X., and Chua, T.-S. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872* (2022).

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

[48] Xia, Y., Liang, Y., Wen, H., Liu, X., Wang, K., Zhou, Z., and Zimmermann, R. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. *arXiv preprint arXiv:2309.13378* (2023).

[49] Yan, H., and Li, Y. A survey of generative ai for intelligent transportation systems. *arXiv preprint arXiv:2312.08248* (2023).

[50] Yang, C., Wu, Q., Wen, Q., Zhou, Z., Sun, L., and Yan, J. Towards out-of-distribution sequential event prediction: A causal treatment. *Advances in neural information processing systems 35* (2022), 22656–22670.

[51] Yang, S., Liu, J., and Zhao, K. Space meets time: Local spacetime neural network for traffic flow forecasting. In *2021 IEEE International Conference on Data Mining (ICDM)* (2021), IEEE, pp. 817–826.

[52] Yi, J., and Park, J. Hypergraph convolutional recurrent neural network. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (2020), pp. 3366–3376.

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

[54] Zhang, G., Yi, J., Yuan, J., Li, Y., and Jin, D. Das: Efficient street view image sampling for urban prediction. *ACM Transactions on Intelligent Systems and Technology 14*, 2 (2023), 1–20.

[55] Zhang, J., Zheng, Y., Qi, D., Li, R., and Yi, X. 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.

[56] Zhang, W., Zhang, L., Han, J., Liu, H., Zhou, J., Mei, Y., and Xiong, H. Irregular traffic time series forecasting based on asynchronous spatio-temporal graph convolutional network. *arXiv preprint arXiv:2308.16818* (2023).
[57] Zhang, Y., Wang, P., Wang, B., Wang, X., Zhao, Z., Zhou, Z., Bai, L., and Wang, Y. Adaptive and interactive multi-level spatio-temporal network for traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems* (2024).
[58] Zhao, T., Liu, Y., Neves, L., Woodford, O., Jiang, M., and Shah, N. Data augmentation for graph neural networks. In *Proceedings of the aaai conference on artificial intelligence* (2021), vol. 35, pp. 11015–11023.
[59] Zhao, Z., Shen, G., Wang, L., and Kong, X. Graph spatial-temporal transformer network for traffic prediction. *Big Data Research* (2024), 100427.
[60] Zheng, C., Fan, X., Wang, C., and Qi, J. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence* (2020), vol. 34, pp. 1234–1241.
[61] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence* (2021), vol. 35, pp. 11106–11115.
[62] Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 4 (2022), 4396–4415.
[63] Zhou, Z., Huang, Q., Wang, B., Hou, J., Yang, K., Liang, Y., and Wang, Y. Coms2t: A complementary spatiotemporal learning system for data-adaptive model evolution. *arXiv preprint arXiv:2403.01738* (2024).
[64] Zhou, Z., Huang, Q., Yang, K., Wang, K., Wang, X., Zhang, Y., Liang, Y., and Wang, Y. Maintaining the status quo: Capturing invariant relations for ood spatiotemporal learning.
[65] Zhou, Z., Yang, K., Liang, Y., Wang, B., Chen, H., and Wang, Y. Predicting collective human mobility via countering spatiotemporal heterogeneity. *IEEE Transactions on Mobile Computing* (2023).

A PSEUDO-CODE OF STONE

We provide the pseudo-code of STONE for spatio-temporal ODD prediction in Algorithm 1.

Algorithm 1: STONE for spatio-temporal ODD prediction

Input: Fréchet embedding $S_{in} \in \mathbb{R}^{N \times d_e}$, observed sampled traffic flow data $\mathbf{x} \in \mathbb{R}^{T \times N \times d}$, ST-module of STONE $\mathcal{F}$ with parameters Θ, masking operators $\mathbb{M} \in \mathbb{R}^{K_M \times N \times N}$

Output: Predicted traffic flow $\hat{Y}$

for $l = 1, 2, ..., L_s$ **do**
  if $l == 1$ **then**
    $S^{(1)} \leftarrow S_{in}$ in Eq. 8; // Shallow encoding
  else
    $S^{(l)} \leftarrow S^{(l-1)}$ in Eq. 9; // Gated transformer
  end
end
$D_t \leftarrow S^{(L_s)}$;
$D_s \leftarrow S_s$ in Eq. 10; // Spatial semantic graph
for $l = 0, 1, 2, ..., L_t$ **do**
  if $l == 0$ **then**
    $X_T^{(0)} \leftarrow$ pos-emb.$(\mathbf{x})$; // Shallow encoding
  else
    $X_T^{(l)} \leftarrow X_T^{(l-1)}$ in Eq. 11; // Dilation TCN
  end
end
$X_t \leftarrow X_T^{(0)}, X_T^{(1)}, ..., X_T^{(L_t)}$ in Eq. 12; // Residual concatenation
$D_t \leftarrow X_t$ in Eq. 13; // Temporal semantic graph
if *Training phase* **then**
  $D_s \leftarrow \mathbb{M} \odot D_s$; // Masking temporal semantic graph
  $D_t \leftarrow \mathbb{M} \odot D_t$; // Masking spatial semantic graph
end
$X_o \leftarrow (D_s, X_t)$ in Eq. 14; // Temporal graph diffusion
$S_o \leftarrow (D_t, S_s)$ in Eq. 14; // Spatial graph diffusion
$\hat{Y} \leftarrow (X_o, S_o)$ in Eq. 15; // Gate decoder

B EXPERIMENT

B.1 Setting

To define the graph topology, we utilized the common practice to construct the normalized geographic adjacency matrix $\mathcal{N}(\mathbf{A}) \in \mathbb{R}^{N \times N}$ for spatio-temporal graph Fréchet embedding computing via the Gaussian kernel [32] with threshold 0.1, whose entries are

$$\mathbf{A}_{uv} = \begin{cases} \exp\left(-\frac{d_{uv}}{\sigma^2}\right), & \text{if } \exp\left(-\frac{d_{uv}}{\sigma^2}\right) > 0.1, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

$$\mathcal{N}(\mathbf{A}) = \frac{\mathbf{A}_{uv}}{\sum_{v \in V} \mathbf{A}_{uv}} \tag{19}$$

Here, d_{uv} denotes the road network distance from sensors u to v, and σ is the standard deviation of all distances.

To optimize performance, we set the batch size to 64 and used the Adam [15] optimizer with a learning rate of $1e-3$ and weight decay of $1e-4$. Additionally, we implemented a learning rate decay strategy where the learning rate is reduced by a factor of 0.95 every 10 training steps, and a gradient clipping strategy with a threshold of 5. The loss function's trade-off parameter, β, is set to 1. The attention network dimension for the adaptive semantic graphs is 20. The TCN comprises 5 hidden layers with a dimension of 128. There are 2 layers of gated transformer, with a hidden dimension of 64 for the SD dataset and 128 for the GBA dataset. The decoder network has a hidden dimension of 128.

B.2 Baseline setting

- **HL** [19] selects the data from the last observation as the predicted value for all future time points.
- **GWNet** [47] has removed the adaptive adjacency matrix due to its lack of scalability. The dimensions for the initialization, skip connections, and output are set to 32, 256, and 512 respectively.
- **STGCN** [53] consists of 2 ST-Blocks. Each ST-Block has two layers of TCN with a dimension of 64 and a kernel size of 3, and a ChebGCN layer with a dimension of 16. The dimension of the output block is 128.
- **STNN** [51] has subgraph-conv with hidden layer dimension 32 and 64.
- **CaST** [48] has removed the random embedding features of nodes, and set the hidden layer dimension to 64, with a granularity of 20 for the environment representation.
- **CauSTG** [64] has removed the node representation. The 4-layer TCNs have (5, 5, 6, 6) kernels with dimension of (12, 6, 3). We set the sub-environment partition number and model number of the sub-environment with 6 and 4 respectively.

B.3 Effect analysis for temporal shift

As shown in Table 5, we report the performance of each model on STSD with only temporal shift. GWNet achieved relatively lower errors, potentially because it inherited diffusion graph convolutional networks, enabling bidirectional modeling of spatio-temporal dependencies and enhancing generalization to spatio-temporal shifts. CaST had better performance than STGCN, because it specifically introduces causal learning to learn invariant patterns for temporal shifts. STONE remained competitive in dealing with temporal shifts. This is because that the proposed graph intervention mechanism can enable the model to learn the invariable pattern efficiently.

Table 5: Prediction performance of each model on datasets with only temporal shift.

ST-SD dataset with only temporal shift									
Model	3 horizon			6 horizon			12 horizon		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HL	29.42	44.26	21.48	51.71	74.81	39.55	93.65	127.43	82.48
STGCN [53]	24.94	38.23	36.93	31.95	47.36	44.09	38.70	55.49	51.25
GWNET [47]	18.38	**29.46**	15.70	25.84	**40.21**	25.31	34.62	53.10	35.70
STNN [51]	39.59	57.73	39.69	39.51	57.42	40.56	42.83	62.04	43.54
CaST [48]	22.83	35.37	27.09	35.45	53.86	41.65	61.00	88.11	70.58
CauSTG [64]	24.97	39.48	43.34	40.04	56.98	47.15	66.83	91.27	75.13
Ours	**18.26**	31.24	**14.95**	**25.06**	40.93	**21.84**	**34.46**	**52.82**	**33.35**

B.4 Discussion

In this section, we discuss the limitations and the future works:

- When there are many random mask operators, the cost of training increases exponentially, and the time it takes to achieve convergence also increases, making it more challenging. This highlights the need for better theories of stochastic optimization. However, this topic is beyond the scope of our work.
- We use gated-Transformers to learn the spatio-temporal semantic graphs. In the future, inspired by the progress of graph structure learning [13], we will design more effective spatio-temporal semantic graph learning models.
- STONE is only experimented on the transportation dataset, and in the future, STONE will be deployed to other spatio-temporal computing domains, such as the atmospheric domain.