ST-FiT: Inductive Spatial-Temporal Forecasting with Limited Training Data

Zhenyu Lei[1], **Yushun Dong**[2], **Jundong Li**[1], **Chen Chen**[1]

[1]University of Virginia
[2]Florida State University
{vjd5zr, jundong, zrh6du}@virginia.edu, yd24f@fsu.edu

Abstract

Spatial-temporal graphs are widely used in a variety of real-world applications. Spatial-Temporal Graph Neural Networks (STGNNs) have emerged as a powerful tool to extract meaningful insights from this data. However, in real-world applications, most nodes may not possess any available temporal data during training. For example, the pandemic dynamics of most cities on a geographical graph may not be available due to the asynchronous nature of outbreaks. Such a phenomenon disagrees with the training requirements of most existing spatial-temporal forecasting methods, which jeopardizes their effectiveness and thus blocks broader deployment. In this paper, we propose to formulate a novel problem of inductive forecasting with limited training data. In particular, given a spatial-temporal graph, we aim to learn a spatial-temporal forecasting model that can be easily generalized onto those nodes without any available temporal training data. To handle this problem, we propose a principled framework named ST-FiT. ST-FiT consists of two key learning components: temporal data augmentation and spatial graph topology learning. With such a design, ST-FiT can be used on top of any existing STGNNs to achieve superior performance on the nodes without training data. Extensive experiments verify the effectiveness of ST-FiT in multiple key perspectives.

Code — https://github.com/LzyFischer/InductiveST
Extended version — https://arxiv.org/abs/2412.10912

Introduction

Spatial-temporal graphs contain both spatial information encoded in graph topology and temporal information encoded in node-associated temporal data (Sahili and Awad 2023). In recent years, spatial-temporal graph data has become ubiquitous in a variety of domains such as transportation (Zhang et al. 2021), epidemiology (Wang et al. 2022), and social science (Kefalas, Symeonidis, and Manolopoulos 2018). In these domains, a widely studied task is spatial-temporal forecasting (Zhang and Patras 2018), i.e., predicting future temporal dynamics associated with the nodes in given spatial-temporal graphs (Ye et al. 2020). Towards such a goal, Spatial-Temporal Graph Neural Networks (STGNNs) stand out (Cui et al. 2021) due to their exceptional capability of synergizing the strengths of the Graph Neural Networks (GNNs) with various sequential forecasting models (Guo et al. 2019; Song et al. 2020) to learn the complex spatial-temporal dependencies. As a consequence, STGNNs have been widely adopted in a plethora of real-world applications (Jin et al. 2023; Zhuang et al. 2022).

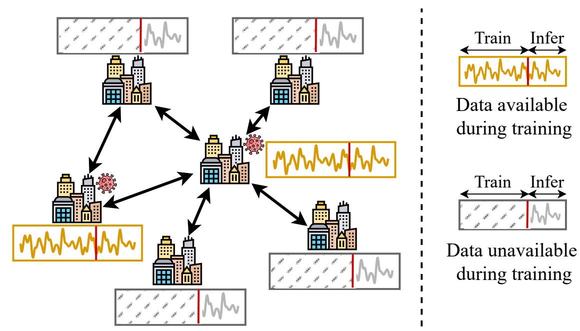


Figure 1: An exemplary spatial-temporal graph where only the temporal data corresponding to a few nodes is accessible during training: on a geographical graph among different cities, only a few cities have available pandemic dynamics at the current time point (marked in red) due to the asynchronous nature of outbreaks. Here, the virus mark denotes the cities that have gone through outbreaks.

Despite advancements, most existing STGNNs require that all nodes in the given spatial-temporal graphs should have temporal data (e.g., time series data) during training (Wang et al. 2020), such that the unique temporal dependency for each node can be easily captured (Shin and Yoon 2024). With the captured temporal dependencies, the STGNNs could make predictions for each node effectively. However, in real-world scenarios, most nodes may not have available temporal data during training (Gupta, Kodamana, and Ranu 2023). We present an exemplary case in Figure 1. Specifically, facing a sudden pandemic such as COVID-19, due to the asynchronous nature of pandemic outbreaks, the pandemic dynamics (e.g., the tendency of confirmed case number) of most cities on a geographical graph may not be available at a given time point (marked in red) (Atchadé and Sokadjo 2022; Panagopoulos, Nikolentzos, and Vazirgiannis 2021). In such cases, existing STGNNs perform poorly in cities without available temporal training data. Therefore, to enable wider deployment, the forecasting model should generalize the learned temporal dependencies to the nodes without any temporal data during training, referred to as *inductive forecasting with limited training data*. A few recent studies have made early explorations to such a problem (Tang et al. 2022). For example, domain adaptation strategies (Fang et al. 2022; Wang et al. 2021) have been adopted to generalize dependencies from nodes with abundant temporal data. However, these works overwhelmingly focus on the generalization between different spatial-temporal graphs, ignoring granular temporal dependency differences within the same graph. Furthermore, they usually require costly fine-tuning (Zhou et al. 2022), which limits their efficiency for real world scenarios (Guo et al. 2023). Therefore, despite the practical significance, the problem of enabling inductive forecasting with limited training data remains underexplored and challenging.

It is worth mentioning that inductive forecasting with limited training data on spatial-temporal graphs presents three challenges. (1) **Limited Temporal Dependencies.** With the temporal data corresponding to only a limited number of nodes available, the forecasting model can only extract limited temporal dependencies (Lachapelle et al. 2024). However, nodes without any temporal training data may still require different temporal dependencies to perform accurate forecasting (Wijsen 2018). Therefore, the first challenge is to learn diversified temporal dependencies for better generalization. (2) **Diverse Spatial Dependencies.** The spatial-temporal graph topology encodes spatial dependencies, crucial for generalizing the learned temporal dependencies between neighboring nodes. However, the topology may exhibit different patterns of spatial dependencies in different local areas (Park and Kim 2014). For example, during the pandemic, geographically neighboring cities may exhibit both similar and distinct pandemic dynamics due to varying interactions (e.g., different volumes of population migration (Gibbs et al. 2020)). Hence, the second challenge is to equip the forecasting model with generalization capability across different spatial dependencies. (3) ***Inference Efficiency.*** Most existing explorations aiming to handle differences in spatial and temporal dependencies require costly fine-tuning processes (Zhou et al. 2022; Ouyang et al. 2022), which makes efficient inference difficult in real-world scenarios. Our third challenge is to avoid additional computational costs and achieve efficient inference on nodes with no available temporal data for training.

To address the challenges above, we introduce ST-FiT (inductive Spatial-Temporal Forecasting with limited Training data), a novel framework that generalizes to different spatial-temporal dependencies without fine-tuning. Specifically, ST-FiT consists of two learning modules for above challenges. To handle the first challenge, ST-FiT introduces a temporal data augmentation module. This module learns the manifold where the available training temporal data lies and generates new temporal data close to it to enrich the training set. In this way, the training temporal dependencies can be enriched , such that the generalization ability to different temporal dependencies can be enhanced for the forecasting model. To handle the second challenge, ST-FiT is equipped with a spatial graph topology learning module. With this module, spatial dependencies (represented as edges in spatial-temporal graphs) between new and existing temporal data can be generated, and existing spatial dependencies can be refined as well. We formulate an optimization problem with an iterative training strategy for these modules. As such, ST-FiT is enabled to perform inductive forecasting with any STGNN backbone while avoiding costly fine-tuning. Such generalization capability and flexibility significantly broaden a broader range of applicable scenarios compared with other alternative spatial-temporal forecasting models. Empirical evaluations on three commonly used real-world datasets corroborate the effectiveness of ST-FiT in multiple key perspectives. Our contributions are summarized as follows:

- **Problem Formulation.** We formulate a novel problem of inductive forecasting with limited training temporal data, which aligns with the setting associated with a wider spectrum of real-world applications.
- **Framework Design.** We design a new framework named ST-FiT to handle the key challenges associated with the problem studied and achieve superior performance compared to other alternatives.
- **Experimental Evaluations.** We conduct comprehensive experiments on three commonly used real-world datasets to verify the effectiveness of our proposed framework.

Problem Definition

In this section, we first present the notations used throughout this paper. Then we introduce two key definitions, including *Spatial-Temporal Graph* and *Spatial-Temporal Forecasting*. Finally, we introduce a novel problem of *Inductive Spatial-Temporal Forecasting with Limited Training Data*.

Notations. We denote an attributed graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A}, \boldsymbol{X})$, where $\mathcal{V}$ is the set of nodes and $N = |\mathcal{V}|$ is the number of nodes. $\boldsymbol{X} \in \mathbb{R}^{N \times C}$ represents the feature matrix of the nodes in $\mathcal{V}$, where C is the dimension number of node features. $\mathcal{E}$ denotes the set of edges, where the edge between node v_i and v_j is denoted as $e_{ij} = (v_i, v_j)$. $\boldsymbol{A} \in \{0, 1\}^{N \times N}$ is the adjacent matrix, where $\boldsymbol{A}_{ij} = 1$ indicates that an edge exits between v_i and v_j, otherwise $\boldsymbol{A}_{ij} = 0$.

Definition 1 ***Spatial-Temporal Graph.*** *A spatial-temporal graph $\{\mathcal{G}^t\}_{t=1}^T$ contains a sequence of graphs $\mathcal{G}^t$ ($1 \leq t \leq T$), where t is the current time step and T is the total number of time steps. Here each $\mathcal{G}^t$ is described as $(\mathcal{V}, \mathcal{E}, \boldsymbol{A}, \boldsymbol{X}^t)$, where $\boldsymbol{X}^t \in \mathbb{R}^{N \times C}$ denotes the node features at time step t. The graph topology described by $\boldsymbol{A}$ reveals the spatial dependency, while the temporal data corresponding to each node consists of all node features across all time steps.*

Problem 1 ***Spatial-Temporal Forecasting*** *Given a sliding window of κ time steps in a spatial-temporal graph $\mathcal{G}^{t-\kappa:t}$, our goal is to learn a function f to predict features $\boldsymbol{X}^{t+1:t+\tau}$ in following τ time steps. Here $(\cdot)^{t_1:t_2}$ denotes a sequence (ordered by time steps) with a length of $t_2 - t_1 + 1$, where the input at time step t_i ($t_1 \leq t_i \leq t_2$) is placed at the ($t_i - t_1 + 1$)-th position.*

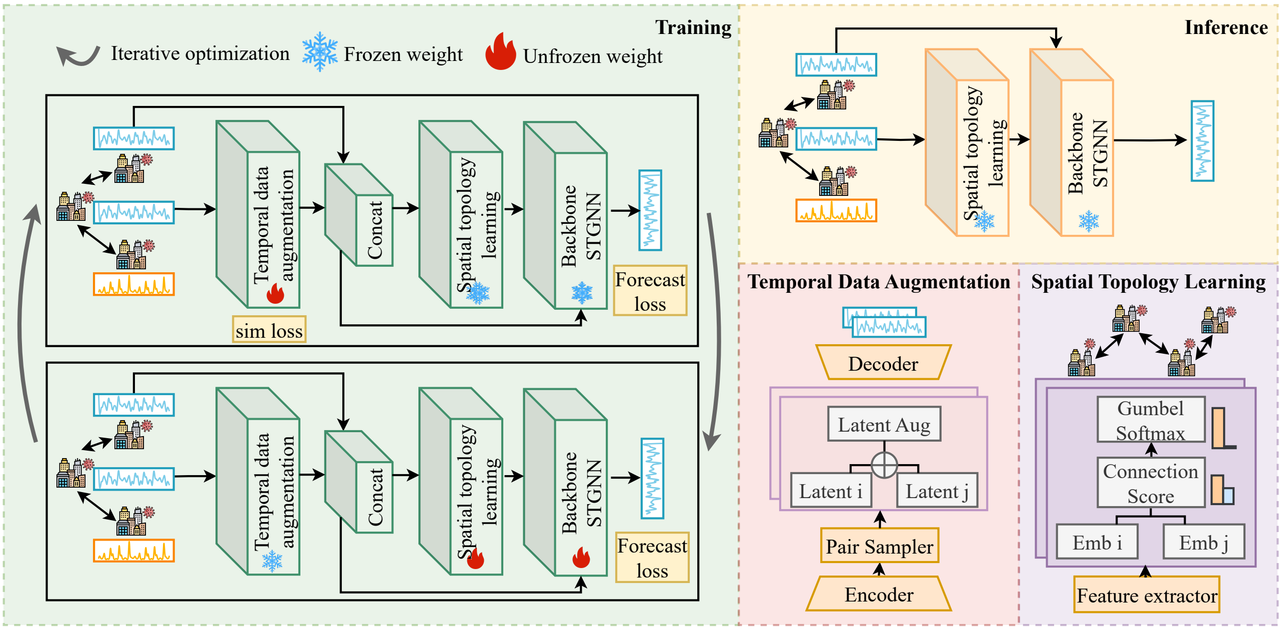


Figure 2: An overview of ST-FiT, including a STGNN backbone, temporal data augmentation, and spatial topology learning.

Based on Definition 1 and Problem 1 above, we then present the problem of *Inductive Spatial-Temporal Forecasting with Limited Training Data* below.

Problem 2 ***Inductive Spatial-Temporal Forecasting with Limited Training Data.*** *Given a spatial-temporal graph with T_{train} time steps $\mathbb{G} = \{\mathcal{G}^t\}_{t=1}^{T_{train}}$ where the features of only a small subset of nodes (denoted as $\mathcal{V}_{train}$ and fixed in all steps) are available, we aim to learn a forecasting model f to accurately predict the whole node feature matrix $\boldsymbol{X}^{T_{train}+1:T_{train}+\tau}$ from step $T_{train}+1$ to $T_{train}+\tau$.*

Methodology

Overview

An overview of ST-FiT is shown in Figure 2. Specifically, ST-FiT consists of 3 modules: (1) *Temporal Data Augmentation* aims to generate diverse temporal data (in the format of time series) via learning the manifold where the accessible training temporal data lies. (2) *Spatial Topology Learning* aims to generate the spatial dependencies between existing and newly generated time series and refine the spatial dependencies between existing time series. (3) *STGNN Backbone*. ST-FiT is plug-and-play, i.e., any STGNN model can be adopted as the backbone of this framework to achieve forecasting on nodes without training temporal data. To effectively optimize the three learning modules, we formulate the optimization problem first and propose an iterative approach to solve it. We introduce the three modules below.

STGNN Backbone

Specifically, an STGNN backbone model takes the temporal data (encoded with temporal dependencies) and the spatial-temporal graph topology (encoded with spatial dependencies) as input, and outputs predicted temporal data. In particular, we assume the temporal data is in the format of time series; the input and output time series are with a length of κ and τ, respectively. We formulate the STGNN backbone as

$$\boldsymbol{x}^{t+1:t+\tau} = h(\boldsymbol{x}^{t-\kappa:t}, \boldsymbol{A}), \tag{1}$$

where h denotes the mapping given by the STGNN backbone; $\boldsymbol{x}^{i:j}$ denotes the time series given by the node feature matrix $\boldsymbol{X}^t$ from time step i to j; $\boldsymbol{x}^{t-\kappa:t}$ and $\boldsymbol{x}^{t+1:t+\tau}$ represent the input and output time series, respectively.

Temporal Data Augmentation

To handle the challenge of limited temporal dependencies, we resort to data augmentation to enrich the training temporal data. However, commonly used methods such as cropping and adding noise in input or latent space are inadequate since they cannot generate new dependencies. In order to achieve our goal, we propose to learn a manifold where all available temporal data lies in the hidden space. Then, we select data points that lie between the points associated with the accessible training temporal data on the manifold, such that new data following new temporal dependencies can be generated. We refer to this strategy as temporal data mix-up on the learned temporal manifold. Specifically, we first input the time series from each sliding window into a Variational Auto-Encoder (VAE) (Kingma and Welling 2013), from which we can derive the representations that characterize the manifold of the temporal data in the hidden space. Denote ξ as a time step between $\kappa+1$ and $T_{train}-\tau$. We achieve the above operations with

$$\begin{aligned} \boldsymbol{\mu}_v, \boldsymbol{\sigma}_v &= \text{Encoder}(\boldsymbol{x}_v^{\xi-\kappa:\xi+\tau}) \text{ and} \\ \boldsymbol{z}_v &= \text{Sample}(\boldsymbol{\mu}_v, \boldsymbol{\sigma}_v), \end{aligned} \tag{2}$$

where v denotes the node (in the given spatial-temporal graph) we are focusing on; $\boldsymbol{\mu}_v$ and $\boldsymbol{\sigma}_v$ are the mean and standard deviation characterizing the Gaussian distribution to be sampled from the latent space; $\kappa+\tau$ denotes the length of the input time series. We assume that $\boldsymbol{z}_v$ from all nodes come from a unified manifold in the hidden space. Then, we can perform temporal data mix-up on the learned manifold. Specifically, we randomly sample K pairs of time series and generate new hidden representations from each pair by

$$\hat{\boldsymbol{z}}_v = \lambda \cdot \boldsymbol{z}_{v_i} + (1-\lambda) \cdot \boldsymbol{z}_{v_j}, \tag{3}$$

where $\mathcal{U}$ is the sampled pair set ($\{v_i, v_j\} \in \mathcal{U}, |\mathcal{U}| = K$); $\hat{\boldsymbol{z}}_v$ is the generated hidden representation; λ is the mix-up ratio. Since the positions for $\boldsymbol{z}_{v_i}$ and $\boldsymbol{z}_{v_j}$ are symmetric, λ naturally falls between 0 and 0.5. We note that this does not rigorously guarantee that $\hat{\boldsymbol{z}}_v$ lies on the learned manifold. However, this generally enables us to generate $\hat{\boldsymbol{z}}_v$ close to the learned manifold, which empirically leads to effective utility improvements. Finally, we transform the generated hidden representations back to input space with a VAE decoder to obtain new temporal data by

$$\hat{\boldsymbol{x}}_v^{\xi-\kappa:\xi+\tau} = \text{Decoder}(\hat{\boldsymbol{z}}_v), \tag{4}$$

where $\hat{\boldsymbol{x}}_v^{\xi-\kappa:\xi+\tau}$ is the generated temporal data.

To optimize the learnable parameters in the encoder and decoder, we design a training objective to maximize the similarity between the available pair $(\boldsymbol{z}_{v_i}, \boldsymbol{z}_{v_j})$ and the newly generated $\hat{\boldsymbol{z}}_v$. The rationale is that the newly generated $\hat{\boldsymbol{z}}_v$ should preserve similar temporal dependencies from both $\boldsymbol{z}_{v_i}$ and $\boldsymbol{z}_{v_j}$. We formulate the optimization goal as

$$\begin{aligned}\mathcal{L}_{sim} = \sum_{\xi} \sum_{\{v_i,v_j\}\in\mathcal{U}} & \lambda \cdot \text{cosine}(g(\{v_i, v_j\}, \xi, \tau), \boldsymbol{z}_{v_i}) \\ & + (1-\lambda) \cdot \text{cosine}(g(\{v_i, v_j\}, \xi, \tau), \boldsymbol{z}_{v_j}),\end{aligned} \tag{5}$$

where function $g(\cdot,\cdot,\cdot)$ denotes the function to generate new time series with node pair (the first parameter), time step (the second parameter), and the steps that go beyond the given time step (the third parameter) through the encoder-decoder design; function $\text{cosine}(\cdot,\cdot)$ takes two vectors as input and outputs their cosine similarity. To further ensure that the generated temporal data with $g(\{v_i, v_j\}, \xi, \tau)$ reflects a consistent dependency across all $\kappa+\tau$ steps, we propose to utilize the backbone STGNN model to perform forecasting on the generated temporal data. Our rationale is that accurate forecasting with the backbone model reveals the existence of a consistent dependency across all time steps. We propose to formulate such an objective as

$$\begin{aligned}\mathcal{L}_{fst} = \sum_{\xi} \sum_{\{v_i,v_j\}\in\mathcal{U}} & \mathcal{L}_{err}(h(g(\{v_i, v_j\}, \xi, \tau)_{[:\xi]}, \boldsymbol{A}), \\ & g(\{v_i, v_j\}, \xi, \tau)_{[-\tau:]}),\end{aligned} \tag{6}$$

where $(\cdot)_{[:\xi]}$ and $(\cdot)_{[-\tau:]}$ denote the first ξ and last τ steps in a given sequence, respectively; function h denotes the backbone STGNN that takes a sequence with a length of ξ and adjacency matrix and outputs the predicted sequence with a length of τ; $\mathcal{L}_{err}$ measures the difference between two input sequences, e.g., the element-wise mean squared error.

Spatial Topology Learning

To handle the challenge of diverse spatial dependencies, we propose to refine existing spatial topology and generate the topology between the generated and existing temporal data. As such, we are able to better adapt the spatial topology to fit the predictive capability of the STGNN backbone. Intuitively, this module should be agnostic to the number of nodes, such that it meets the need for inductive forecasting. Meanwhile, the learned spatial topology should be naturally discrete and sparse so that it only encodes key patterns of spatial dependencies (Hu et al. 2022). To achieve the goals above, we propose to leverage the Gumbel-Softmax reparameterization to learn a sparse graph topology based on the node features (Shang, Chen, and Bi 2021).

Specifically, we first use a Multi-Layer Perceptron (MLP) encoder to transform each sliding window of time series into a hidden representation. Then, we use another MLP maps the hidden representations for each pair of nodes v_i, v_j to a scalar $\boldsymbol{P}_{ij} \in [0, 1]$. We utilize the matrix $\boldsymbol{P}$ to parameterize a Bernoulli distribution between every node pair. By drawing samples from the Bernoulli distributions, we are able to construct a refined adjacency matrix $\tilde{\boldsymbol{A}}$ to characterize the learned spatial topology, i.e., $\tilde{\boldsymbol{A}}_{ij} \sim Ber(\boldsymbol{P}_{ij})$. Note that we apply the Gumbel reparameterization trick (Franceschi et al. 2019) to enable the gradient to flow through $\tilde{\boldsymbol{A}}$, such that gradient-based techniques can be adopted to optimize $\boldsymbol{P}$. We formulate the procedure to derive $\tilde{\boldsymbol{A}}$ as

$$\tilde{\boldsymbol{A}}_{ij} = \text{Gumbel-Softmax}(\boldsymbol{P}_{ij}, s), \tag{7}$$

where s is the temperature parameter of Gumbel-Softmax. However, based on the formulation given above, it becomes difficult then to impose l_1 norm as a regularization to achieve sparse graph topology. To enforce the learned spatial topology to be sparse, we propose to transform the learned matrix $\boldsymbol{P}_{ij}$ with a threshold ϵ, i.e.,

$$\hat{\boldsymbol{P}}_{ij} = \text{Sigmoid}(\mathrm{e}^{(\boldsymbol{P}_{ij}-\epsilon)/\phi}), \tag{8}$$

where ϕ is the temperature. Intuitively, $\boldsymbol{P}_{ij} < \epsilon$ will make it less likely to generate an edge between node v_i and v_j.

Optimization Strategy and Inference

The training objectives of ST-FiT are three-fold: (1) Generate diverse temporal data that lies close to the learned manifold; (2) Refine spatial topology based on diverse spatial dependencies; (3) Capture key spatial-temporal dependencies for forecasting with the STGNN backbone. However, we note that the optimization of temporal data augmentation and the other two modules are intertwined, since (1) it requires the STGNN backbone to perform prediction ; and (2) it requires the refined spatial dependencies from the spatial topology learning module as the input (as in $\mathcal{L}_{fst}$). As such, we propose to formulate the overall optimization problem and solve it with an iterative training strategy, i.e., training the temporal data augmentation module and the other two modules iteratively. We refer to the two optimization processes in each iteration as the *Phase 1* (optimizing the temporal data augmentation module) and the *Phase 2* (optimizing other modules), respectively.

In *Phase 1*, we optimize the temporal data augmentation with gradient-based optimization techniques, while the parameters of other modules are frozen. Formally, we have

$$\boldsymbol{\theta}^{aug}_{\text{epoch+1}} = \boldsymbol{\theta}^{aug}_{\text{epoch}} - \eta \cdot \nabla_{\boldsymbol{\theta}^{aug}} \mathcal{L}_{aug}, \tag{9}$$

$$\mathcal{L}_{aug} = \mathcal{L}_{sim} + \mathcal{L}_{fst} + \mathcal{L}_{KL}, \tag{10}$$

where $\boldsymbol{\theta}^{aug}$ denotes learnable parameters for temporal data augmentation module and $\mathcal{L}_{KL}$ denotes the commonly used regularization term for the adopted VAE (Verma et al. 2019).

In *Phase 2*, we aim to jointly optimize the learnable parameters in the STGNN backbone and the spatial topology learning module, where the parameters of temporal data augmentation are frozen. Formally, we have

$$\boldsymbol{\theta}^{gf}_{\text{epoch + 1}} = \boldsymbol{\theta}^{gf}_{\text{epoch}} - \eta \cdot \nabla_{\boldsymbol{\theta}^{gf}} \mathcal{L}_{gf}, \tag{11}$$

$$\mathcal{L}_{gf} = \mathcal{L}_{fst} + \mathcal{L}_{ori} \tag{12}$$

where $\boldsymbol{\theta}^{gf}$ denotes the learnable parameters associated with the STGNN backbone and spatial topology learning module. $\mathcal{L}_{fst}$ and $\mathcal{L}_{ori}$ are loss functions of the forecasting results for the generated and original time series, respectively. In this way, the parameters in the spatial topology learning module and the STGNN backbone are jointly optimized.

Finally, during inference, we propose to sample a graph topology characterized by $\tilde{\boldsymbol{A}}$ based on the learned $\boldsymbol{P}$. As such, we are able to directly perform forecasting. We present the complete algorithmic routine in Appendix.

Experimental Evaluations

In this section, we aim to answer the following research questions. **RQ1.** How well can ST-FiT generalize to nodes with no available temporal data for training, compared to other existing alternatives? **RQ2.** How does the performance tendency of ST-FiT look like compared with other baselines when these models are trained on varying ratios of nodes with available temporal data? **RQ3.** How does each module of ST-FiT contribute to the overall performance? **RQ4.** How does the choice of hyper-parameters influence the performance of ST-FiT? In the following sections, we first present the experimental settings, followed by the answers to the proposed research questions.

Experimental Settings

Below we provide a brief introduction to the experiment settings, the details will be explained in Appendix.

Datasets. Following previous works (Li et al. 2023), we conduct experiments on three most commonly used real-world datasets PEMS03, PEMS04, and PEMS08, which are all public transport network datasets released by Caltrans Performance Measurement System (PeMS) (PeMS 2021).

Baselines. Since the studied setting is novel, which requires the model to be inductive, we compare our framework to state-of-the-art baselines applicable to such experimental setting. **Linear Sum:** *(1) Historical Average (HA)* (Dai et al. 2020). **Temporal-based**: *(2) FC-LSTM* (Sutskever and Vinyals 2014). **Spatial-Temporal**: *(3) STGCN* (Yu, Yin, and Zhu 2017). *(4) STGODE* (Fang et al. 2021). **Fine-tuning:** *(5) TransGTR* (Jin, Chen, and Yang 2023).

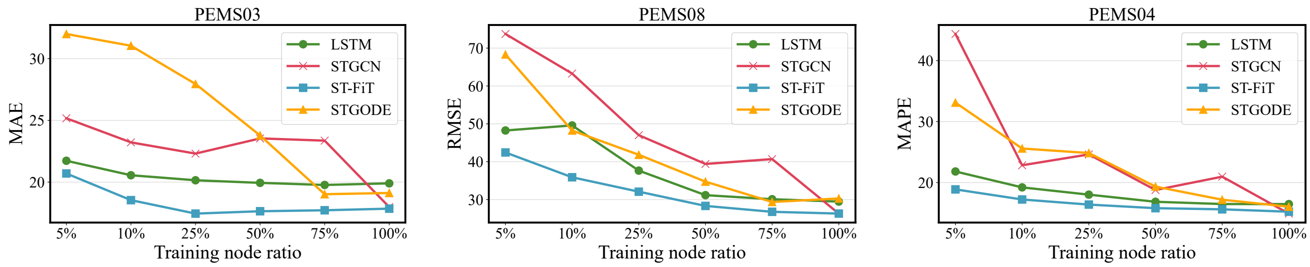
Datasets	Methods	MAE	RMSE	MAPE (%)
PEMS04	HA	41.98 (±0.00)	61.50 (±0.00)	29.92 (±0.00)
	FC-LSTM	<u>28.17</u> (±0.32)	<u>44.38</u> (±0.46)	<u>19.21</u> (±0.38)
	STGODE	34.35 (±1.62)	51.54 (±1.98)	25.59 (±3.03)
	STGCN	32.60 (±0.20)	48.89 (±0.74)	23.40 (±0.58)
	TransGTR	32.76 (±2.30)	48.94 (±5.86)	26.87 (±2.54)
	ST-FiT	**25.11** (±0.42)	**39.30** (±0.62)	**17.23** (±0.43)
PEMS03	HA	32.47 (±0.00)	49.80 (±0.00)	30.59 (±0.00)
	FC-LSTM	20.56 (±0.06)	33.96 (±0.36)	20.41 (±0.40)
	STGODE	31.05 (±1.75)	53.23 (±7.95)	30.20 (±1.26)
	STGCN	23.22 (±0.24)	37.70 (±1.15)	22.91 (±0.80)
	TransGTR	**17.50** (±0.78)	**28.35** (±1.16)	**18.11** (±0.75)
	ST-FiT	<u>18.40</u> (±0.23)	<u>29.31</u> (±0.32)	<u>18.94</u> (±1.53)
PEMS08	HA	34.56 (±0.00)	50.41 (±0.00)	21.60 (±0.00)
	FC-LSTM	30.52 (±0.78)	49.58 (±1.74)	16.33 (±0.30)
	STGODE	30.00 (±2.10)	48.24 (±5.23)	18.62 (±0.55)
	STGCN	41.67 (±1.25)	63.48 (±1.37)	33.48 (±3.80)
	TransGTR	**18.00** (±0.61)	**28.32** (±0.79)	**11.30** (±0.99)
	ST-FiT	<u>25.09</u> (±0.18)	<u>39.52</u> (±0.46)	<u>14.48</u> (±0.47)

Table 1: Average performance of forecasting. The best results and the second best results are in bold and underlined, respectively. All experiments have been repeated with 3 different random seeds. ST-FiT outperforms baselines without fine-tuning on all datasets, and achieves competitive performance with fine-tuning baseline TransGTR.

Task Settings. For a fair comparison, we follow the dataset division along temporal dimensions in previous works (Jiang et al. 2023a), where datasets are split as 70% training, 20% validation and 10% inference in chronological order. For the task setting of inductive forecasting with limited training data, we randomly choose the temporal data from 10% nodes for training, and adopt the same split for validation. Following previous works, we generate training samples through a sliding window of 24 time steps, with the first 12 as model input, and the remaining 12 as ground truth for forecasting outcomes. Accordingly, we compare the average performance on the MAE, RMSE, and MAPE metrics.

Generalization Performance

To answer **RQ1**, we first evaluate the forecasting performance of ST-FiT on those nodes without available temporal data during training. We make following observations from the the empirical results in Table **??**. (1) ST-FiT outperforms all baselines that do not require fine-tuning. Notably, ST-FiT exceeds the associated backbone model STGCN by up to 40.0% in MAE, 38.6% in RMSE, and 55.6% in MAPE. This verifies the effectiveness of ST-FiT in generalizing to the nodes with different temporal dependencies. Meanwhile, ST-FiT also significantly outperforms FC-LSTM and STGODE, which further demonstrates its superiority over different types of state-of-the-art alternatives. (2) ST-FiT achieves comparable performance with fine-tuned model TransGTR. Specifically, ST-FiT outperforms TransGTR on PEMS04 by 27.1% in MAE, 19.0% in RMSE, and 34.8% in MAPE. In addition, ST-FiT has comparable performances



(a) MAE for PEMS03

(b) RMSE for PEMS08

(c) MAPE for PEMS04

Figure 3: The performance of ST-FiT compared to baselines with different training node ratios. As training node ratio decreases, the performances of all models drops, while ST-FiT outperforms other baselines over all ratios consistently.

with TransGTR on PEMS03. We note that compared with ST-FiT, TransGTR adopts an additional fine-tuning process with much more abundant temporal data. As such, TransGTR can extract temporal data from all nodes in the fine-tuning process, which helps it capture more spatial-temporal dependencies. As a result, we argue that achieving competitive performance to TransGTR without relying on any fine-tuning process should be regarded as satisfactory performance, and this further corroborates the effectiveness of ST-FiT in capturing diverse spatial-temporal dependencies.

Performance w.r.t. Training Node Ratios

To answer **RQ2**, we train the proposed ST-FiT with different ratios of nodes whose temporal data is available during training. We compare its performance tendencies with those from baselines without fine-tuning. We select a wide range of ratios including 5%, 10%, 25%, 50%, 75%, and 100% to test whether ST-FiT can outperform other aternatives consistently w.r.t. different ratios. The results are shown in Figure 3. We make the following observations: (1) ST-FiT consistently outperforms baselines on all ratios, which demonstrates its effectiveness under different degrees of training data limitation. (2) ST-FiT performs better with fewer training nodes. For PEMS08, we could observe 28.1% improvement in RMSE compared to STGCN with 50% ratio, while it increases to 42.4% with 5% ratio. This observation demonstrates ST-FiT's effectiveness especially when training with severely limited data. (3) When all nodes have temporal data for training, ST-FiT still remains comparable with the best baselines, which indicates our temporal data augmentation and spatial topology learning do not impair the STGNN backbone's original performance and are widely applicable.

Ablation Study

To answer **RQ3**, we investigate how our proposed modules contribute to the forecasting performance separately. We use *w/o aug* to denote cases without temporal data augmentation module. *W/o gl* denotes removing the spatial topology learning module. Furthermore, we design two experiments for temporal data augmentation, where *w/o sim* denotes removing $\mathcal{L}_{sim}$, and *w/o fst* denotes removing $\mathcal{L}_{fst}$. Additionally, we design two variants of our proposed spatial topology learning module. specifically, *w/o gs* denotes adopting a

Variants	MAE	RMSE	MAPE (%)
ST-FiT	**25.11** (±0.42)	**39.30** (±0.62)	**17.23** (±0.43)
w/o aug	27.31 (±0.18)	43.04 (±0.29)	18.64 (±0.10)
w/o sim	<u>25.55</u> (±0.14)	<u>40.18</u> (±0.44)	<u>17.41</u> (±0.16)
w/o fst	26.61 (±1.06)	41.66 (±1.61)	19.00 (±0.56)
w/o gl	27.58 (±0.30)	43.50 (±0.55)	19.00 (±0.28)
w/o gs	26.68 (±1.12)	42.55 (±2.20)	17.76 (±0.35)
identity	28.00 (±0.18)	44.48 (±0.36)	19.38 (±0.60)

Table 2: Performance comparison of forecasting for ablation study. The best results are in bold, and the second best results are underlined. It is observed that removing any module of ST-FiT will jeopardize the overall performance.

fully connected graph. *Identity* denotes replacing $\boldsymbol{A}$ with an identity matrix. The results are shown in Table **??**. We make the observations as follows: (1) Both modules contribute to the overall performance, which verifies their effectiveness for improving generalization. (2) Removing $\mathcal{L}_{sim}$ or $\mathcal{L}_{fst}$ degrades the performance, which indicates their ability in generating temporal data with consistent temporal dependencies. (3) Spatial topology learning performs better than variant, which indicates the learned spatial topology significantly enhances the diversity of spatial dependencies.

Parameter Sensitivity

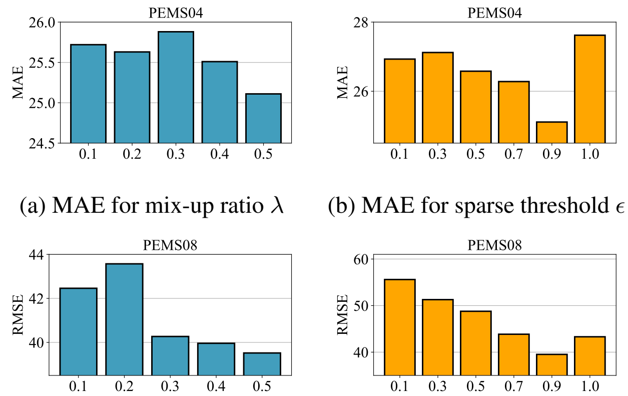
To answer **RQ4**, we analyze the impact of hyper-parameter values, including the threshold in controlling sparsity of spatial topology learning ϵ and the parameter λ for temporal data mix-up. We choose values of ϵ from 0 to 1, where higher value denotes a sparser learned graph topology. For the value of λ, we choose it from the range between 0 and 0.5. The results of three datasets are shown in Figure 4. For the impact of the sparsity of the graph, we observe that: (1) Sparser structures generally perform better, which indicates the necessity of sparsity in spatial-temporal forecasting on nodes without available training temporal data. (2) Overly sparse structures harm performance, since it can omit certain key connections between nodes. From Figure 4a and 4b, we are able to observe that the higher value of λ improve the performance, which can be attributed to a higher diversity of the generated temporal data. With above experiments, we recommend using λ as 0.5, ϵ as 0.9 for optimal performance.

Related Works

Spatial-Temporal Forecasting. Spatial-temporal forecasting is crucial but challenging. STGNNs have shown promise in this area but typically require temporal data for all nodes during training, making them unsuitable for inductive forecasting (Deng et al. 2021). Some other STGNNs can perform inductive forecasting (Wu et al. 2019; Jiang et al. 2023b) but struggle with limited training data, where they only extract limited spatial-temporal dependencies and have difficulty adapting to temporal data with new dependencies. Recently, several works have resorted to domain adaptation for such generalization challenges. They managed to extract temporal dependencies from nodes without temporal data during training, which offers a possible solution to inductive forecasting (Cheng et al. 2023; Ouyang et al. 2023) on limited training data. However, these works only focus on graph-level generalization and require costly fine-tuning. Differently, ST-FiT enriches the training data with diverse temporal dependencies through data augmentation and captures different spatial dependencies between existing and new temporal data, improving generalization.

Temporal Data Augmentation. Due to the common scarcity of temporal data in real-world scenarios, existing works propose to generate temporal data with data augmentation techniques (Fu, Kirchbuchner, and Kuijper 2020). The most challenging part is to generate data not only with diverse temporal dependencies but also lying close to the manifold of existing temporal data. Most traditional algorithms such as slicing, jittering, or scaling apply simple transformation and perturbation (Um et al. 2017; Iwana and Uchida 2021), where they either fail to generate temporal data with diverse and consistent temporal dependencies. Recent prevailing deep learning models such as Generative Adversarial Networks (GANs) (Goodfellow et al. 2020) and Variational Autoencoders (VAEs) (Goubeaud et al. 2021) are promising solutions to generating more consistent temporal data that lie close to the manifold of existing temporal data. Nevertheless, the generated temporal data lack diversity, which limits the contribution to generalization ability. In this work, we present ST-FiT, which adopts mix-up on the manifold and learns to capture the temporal dependencies. This helps to not only enrich the original latent space region due to mix-up on the manifold (Huh et al. 2024), but also ensure that the generated temporal data lies close to the manifold where the available training temporal data lies.

Graph Topology Learning. Due to incompleteness and scarcity of existing graph topology, substantial work have devoted to learning better graph topology over diverse types of network data such as brain networks (Cui et al. 2022), and social networks (Zhang et al. 2022). One of the most important problems is to learn a sparse and discrete graph topology that not only represents the real-world scenarios but also contains few spurious connections (Jin et al. 2020). (Franceschi et al. 2019) proposed to leverage Gumbel-Softmax reparameterization tricks in learning discrete structures with bilevel optimization. (Shang, Chen, and Bi 2021) further applied Gumbel-Softmax reparameterization to spatial-temporal forecasting with uni-level optimization. However, they both require additional resources such as a pre-defined KNN to achieve topology sparsity. In contrast, ST-FiT employs a simple transformation to attain any level of sparsity without increasing the computational burden.



(a) MAE for mix-up ratio λ (b) MAE for sparse threshold ϵ

(c) RMSE for mix-up ratio λ (d) RMSE for sparse threshold ϵ

Figure 4: Performance of ST-FiT with different mix-up ratios λ and sparse thresholds $\emptyset$. The mix-up ratio behaves slight influence, while the positive correlation between mix-up ratio λ and the performance still exhibits. Sparsity has positive correlation with performance, while extreme sparsity brings negative influence due to loss of key connections.

Conclusion

In this paper, we study an under-explored research problem of inductive forecasting with limited training data, which requires models to generalize the learned spatial-temporal dependencies from the nodes with available training temporal data to those nodes without. To handle this problem, we propose ST-FiT that can achieve superior performance without additional fine-tuning. Overall, two key learning modules contribute to the superiority of ST-FiT. Specifically, the temporal data augmentation allows us to generate diverse temporal data which lies close to the manifold of the training temporal data. Hence the model is enabled to generalize to nodes with different temporal dependencies. Meanwhile, the spatial topology learning refines the spatial dependencies, which improves adaptation to the backbone STGNNs to achieve better forecasting performance. We also propose an iterative training strategy for the optimization of these modules. Extensive experiments on three commonly used real-world datasets verify the effectiveness of ST-FiT.

Acknowledgments

This work is supported in part by the National Science Foundation under grants IIS-2006844, IIS-2144209, IIS-2223769, IIS-2331315, CNS-2154962, BCS-2228534, and CMMI-2411248, Office of Naval Research under grant N000142412636, the Commonwealth Cyber Initiative Awards under grants VV-1Q24-011, VV-1Q25-004, and the research gift funding from Netflix and Snap.

References

Atchadé, M. N.; and Sokadjo, Y. M. 2022. Overview and cross-validation of COVID-19 forecasting univariate models. *Alexandria Engineering Journal*, 61(4): 3021–3036.

Cheng, Y.; Chen, P.; Guo, C.; Zhao, K.; Wen, Q.; Yang, B.; and Jensen, C. S. 2023. Weakly guided adaptation for robust time series forecasting. *Proceedings of the VLDB Endowment*, 17(4): 766–779.

Cui, H.; Dai, W.; Zhu, Y.; Kan, X.; Gu, A. A. C.; Lukemire, J.; Zhan, L.; He, L.; Guo, Y.; and Yang, C. 2022. Braingb: a benchmark for brain network analysis with graph neural networks. *IEEE transactions on medical imaging*, 42(2): 493–506.

Cui, Y.; Zheng, K.; Cui, D.; Xie, J.; Deng, L.; Huang, F.; and Zhou, X. 2021. METRO: a generic graph neural network framework for multivariate time series forecasting. *Proceedings of the VLDB Endowment*, 15(2): 224–236.

Dai, R.; Xu, S.; Gu, Q.; Ji, C.; and Liu, K. 2020. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, 3074–3082.

Deng, J.; Chen, X.; Jiang, R.; Song, X.; and Tsang, I. W. 2021. 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*, 269–278.

Fang, Z.; Long, Q.; Song, G.; and Xie, K. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 364–373.

Fang, Z.; Wu, D.; Pan, L.; Chen, L.; and Gao, Y. 2022. When Transfer Learning Meets Cross-City Urban Flow Prediction: Spatio-Temporal Adaptation Matters. In *IJCAI*, volume 22, 2030–2036.

Franceschi, L.; Niepert, M.; Pontil, M.; and He, X. 2019. Learning discrete structures for graph neural networks. In *International conference on machine learning*, 1972–1982. PMLR.

Fu, B.; Kirchbuchner, F.; and Kuijper, A. 2020. Data augmentation for time series: traditional vs generative models on capacitive proximity time series. In *Proceedings of the 13th ACM international conference on pervasive technologies related to assistive environments*, 1–10.

Gibbs, H.; Liu, Y.; Pearson, C. A.; Jarvis, C. I.; Grundy, C.; Quilty, B. J.; Diamond, C.; and Eggo, R. M. 2020. Changing travel patterns in China during the early stages of the COVID-19 pandemic. *Nature communications*, 11(1): 5012.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Goubeaud, M.; Joußen, P.; Gmyrek, N.; Ghorban, F.; Schelkes, L.; and Kummert, A. 2021. Using variational autoencoder to augment sparse time series datasets. In *2021 7th international conference on optimization and applications (ICOA)*, 1–6. IEEE.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 922–929.

Guo, S.; Lin, Y.; Gong, L.; Wang, C.; Zhou, Z.; Shen, Z.; Huang, Y.; and Wan, H. 2023. Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 1585–1596. IEEE.

Gupta, M.; Kodamana, H.; and Ranu, S. 2023. Frigate: Frugal spatio-temporal forecasting on road networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 649–660.

Hu, Z.; Zhao, Z.; Yi, X.; Yao, T.; Hong, L.; Sun, Y.; and Chi, E. 2022. Improving multi-task generalization via regularizing spurious correlation. *Advances in Neural Information Processing Systems*, 35: 11450–11466.

Huh, I.; Choe, J. M.; KIM, Y.; Kim, D.; et al. 2024. Isometric Quotient Variational Auto-Encoders for Structure-Preserving Representation Learning. *Advances in Neural Information Processing Systems*, 36.

Iwana, B. K.; and Uchida, S. 2021. Time series data augmentation for neural networks by time warping with a discriminative teacher. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 3558–3565. IEEE.

Jiang, J.; Wu, B.; Chen, L.; Zhang, K.; and Kim, S. 2023a. Enhancing the Robustness via Adversarial Learning and Joint Spatial-Temporal Embeddings in Traffic Forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 987–996.

Jiang, R.; Wang, Z.; Yong, J.; Jeph, P.; Chen, Q.; Kobayashi, Y.; Song, X.; Fukushima, S.; and Suzumura, T. 2023b. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8078–8086.

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

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

Jin, Y.; Chen, K.; and Yang, Q. 2023. 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*, 1032–1043.

Kefalas, P.; Symeonidis, P.; and Manolopoulos, Y. 2018. Recommendations based on a heterogeneous spatio-temporal social network. *World Wide Web*, 21: 345–371.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lachapelle, S.; López, P. R.; Sharma, Y.; Everett, K.; Priol, R. L.; Lacoste, A.; and Lacoste-Julien, S. 2024. Nonparametric Partial Disentanglement via Mechanism Sparsity: Sparse Actions, Interventions and Sparse Temporal Dependencies. *arXiv preprint arXiv:2401.04890*.

Li, H.; Jin, D.; Li, X.; Huang, J.; Ma, X.; Cui, J.; Huang, D.; Qiao, S.; and Yoo, J. 2023. Dmgf-net: an efficient dynamic multi-graph fusion network for traffic prediction. *ACM Transactions on Knowledge Discovery from Data*, 17(7): 1–19.

Ouyang, C.; Chen, C.; Li, S.; Li, Z.; Qin, C.; Bai, W.; and Rueckert, D. 2022. Causality-inspired single-source domain generalization for medical image segmentation. *IEEE Transactions on Medical Imaging*, 42(4): 1095–1106.

Ouyang, X.; Yang, Y.; Zhou, W.; Zhang, Y.; Wang, H.; and Huang, W. 2023. Citytrans: Domain-adversarial training with knowledge transfer for spatio-temporal prediction across cities. *IEEE Transactions on Knowledge and Data Engineering*.

Panagopoulos, G.; Nikolentzos, G.; and Vazirgiannis, M. 2021. Transfer graph neural networks for pandemic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4838–4845.

Park, Y. M.; and Kim, Y. 2014. A spatially filtered multilevel model to account for spatial dependency: application to self-rated health status in South Korea. *International journal of health geographics*, 13: 1–10.

PeMS, C. 2021. Caltrans performance measurement system (pems). *US DOT*.

Sahili, Z. A.; and Awad, M. 2023. Spatio-temporal graph neural networks: A survey. *arXiv preprint arXiv:2301.10569*.

Shang, C.; Chen, J.; and Bi, J. 2021. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*.

Shin, Y.; and Yoon, Y. 2024. PGCN: Progressive graph convolutional networks for spatial–temporal traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems*.

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

Sutskever, I.; and Vinyals, O. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*.

Tang, Y.; Qu, A.; Chow, A. H.; Lam, W. H.; Wong, S. C.; and Ma, W. 2022. Domain adversarial spatial-temporal network: A transferable framework for short-term traffic forecasting across cities. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1905–1915.

Um, T. T.; Pfister, F. M.; Pichler, D.; Endo, S.; Lang, M.; Hirche, S.; Fietzek, U.; and Kulić, D. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*, 216–220.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, 6438–6447. PMLR.

Wang, L.; Adiga, A.; Chen, J.; Sadilek, A.; Venkatramanan, S.; and Marathe, M. 2022. Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 12191–12199.

Wang, S.; Miao, H.; Li, J.; and Cao, J. 2021. Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(5): 4695–4705.

Wang, X.; Ma, Y.; Wang, Y.; Jin, W.; Wang, X.; Tang, J.; Jia, C.; and Yu, J. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*, 1082–1092.

Wijsen, J. 2018. Temporal Dependencies.

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

Ye, J.; Zhao, J.; Ye, K.; and Xu, C. 2020. Multi-stgcnet: A graph convolution based spatial-temporal framework for subway passenger flow forecasting. In *2020 International joint conference on neural networks (IJCNN)*, 1–8. IEEE.

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

Zhang, C.; and Patras, P. 2018. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 231–240.

Zhang, X.; Huang, C.; Xu, Y.; Xia, L.; Dai, P.; Bo, L.; Zhang, J.; and Zheng, Y. 2021. Traffic flow forecasting with spatial-temporal graph diffusion network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 15008–15015.

Zhang, Y.; Gao, H.; Pei, J.; and Huang, H. 2022. Robust self-supervised structural graph neural network for social network prediction. In *Proceedings of the ACM Web Conference 2022*, 1352–1361.

Zhou, K.; Liu, Z.; Qiao, Y.; Xiang, T.; and Loy, C. C. 2022. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhuang, D.; Wang, S.; Koutsopoulos, H.; and Zhao, J. 2022. Uncertainty quantification of sparse travel demand prediction with spatial-temporal graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4639–4647.