FlashST: A Simple and Universal Prompt-Tuning Framework for Traffic Prediction

Zhonghang Li [1 2] **Lianghao Xia** [2] **Yong Xu** [1 3] **Chao Huang*** [2]

Abstract

The objective of traffic prediction is to accurately forecast and analyze the dynamics of transportation patterns, considering both space and time. However, the presence of distribution shift poses a significant challenge in this field, as existing models struggle to generalize well when faced with test data that significantly differs from the training distribution. To tackle this issue, this paper introduces a simple and universal spatio-temporal prompt-tuning framework-FlashST, which adapts pre-trained models to the specific characteristics of diverse downstream datasets, improving generalization in diverse traffic prediction scenarios. Specifically, the FlashST framework employs a lightweight spatio-temporal prompt network for in-context learning, capturing spatio-temporal invariant knowledge and facilitating effective adaptation to diverse scenarios. Additionally, we incorporate a distribution mapping mechanism to align the data distributions of pre-training and downstream data, facilitating effective knowledge transfer in spatio-temporal forecasting. Empirical evaluations demonstrate the effectiveness of our FlashST across different spatio-temporal prediction tasks using diverse urban datasets. Code is available at https://github.com/HKUDS/FlashST.

1. Introduction

Accurate traffic forecasting is a crucial objective in the domain of urban computing. It aims to precisely predict and analyze the dynamic transportation patterns within cities. The key goals of traffic prediction include supporting urban planning, enabling real-time monitoring and management of traffic, and contributing to the development of smart city applications (Guo et al., 2019; Han et al., 2021; Li et al., 2024).. The primary challenge lies in effectively modeling the intricate spatial and temporal correlations among different time intervals and geographical locations.

Neural network techniques have become a prominent approach for spatio-temporal prediction tasks, thanks to their remarkable capabilities in feature representation. Initially, researchers leveraged Recurrent Neural Networks (RNNs) (Yu et al., 2017) to capture temporal dependencies, and Convolutional Neural Networks (CNNs) (Zhang et al., 2017; Yao et al., 2018) to model spatial correlations. Subsequently, Graph Neural Networks (GNNs) have emerged as effective models for capturing complex spatial dependencies, utilizing message passing mechanisms to learn the relationships among spatial units such as regions and road segments. In constructing the graph adjacency matrices, researchers have explored various factors. Some have considered static geographical distance (Li et al., 2018; Yu et al., 2018), while others have incorporated time-aware region correlations (Han et al., 2021; Li et al., 2022). Additionally, there have been efforts to learn the region-wise relevance in a data-driven manner (Wu et al., 2019; 2020a).

Although the aforementioned methods have shown effectiveness, most current spatio-temporal prediction models struggle to generalize effectively when confronted with distribution shifts across diverse downstream datasets and tasks. In their methods, the assumption of an inconsistent distribution between training and testing data becomes a hindrance to accurate predictions in real-life urban scenarios with uncontrolled shifts from unseen spatio-temporal data. As illustrated in Figure 1, directly applying parameters learned from training on dataset A to test on dataset B can lead to suboptimal performance due to significant variations in spatio-temporal characteristics across different data distributions. Therefore, there is a need to enhance the generalization ability of spatio-temporal forecasting models by efficiently adapting them to handle such distribution shifts.

While there are potential benefits to enabling model adaptation for spatio-temporal prediction methods, several key challenges remain to be addressed. *Firstly*, **C1**: efficiently distilling specific and complex spatio-temporal contextual information from the downstream tasks is essential. How-

[1]South China University of Technology [2]University of Hong Kong [3]PAZHOU LAB. Correspondence to: Chao Huang <chaohuang75@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

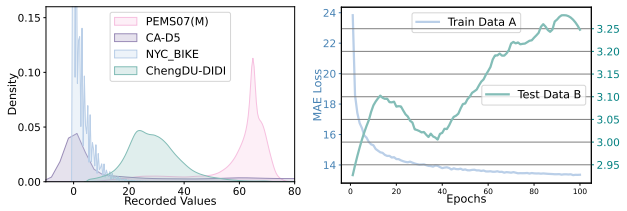


Figure 1. Motivations behind FlashST. The left figure illustrates the diverse data distributions across various ST datasets, while the right figure demonstrates that the end-to-end model's parameters are overfit to training set A and fail to generalize to test set B.

ever, equipping pre-trained models with the ability to understand and incorporate the spatial and temporal characteristics of new domain data that is only accessible during testing is a formidable challenge. *Secondly*, **C2**: there is often a significant distribution gap between the training and test datasets, especially when they are collected from different spatio-temporal scenarios and domains. Consequently, it is essential to enhance the model adaptation framework by enabling it to effectively bridge the distribution gap and capture spatio-temporal invariants. This enhancement will facilitate the successful knowledge transfer from the pre-trained phase to the downstream prediction tasks.

Contributions. To overcome the challenges mentioned above, we introduce a lightweight and innovative prompt-tuning framework (FlashST) that aims to achieve spatio-temporal in-context learning, enabling efficient and effective model adaptation across different spatio-temporal prediction tasks. Within our framework, we employ a spatio-temporal prompt network to enable in-context learning. To tackle challenge **C1**, we propose a context distillation mechanism that captures contextual signals from unseen data, facilitating adaptation to diverse spatio-temporal scenarios. Furthermore, a dependency modeling scheme is introduced to capture and understand the relationships between time and locations, enabling effective analysis of inter-dependencies among spatio-temporal elements. To overcome challenge **C2**, we enhance our FlashST framework by incorporating a unified distribution mapping mechanism, which bridges the distribution gap between pre-trained and downstream tasks. This mechanism facilitates the effective transfer of knowledge from the pre-trained insights to downstream spatio-temporal forecasting by aligning the data distributions through the regularization of prompt embeddings.

We conduct extensive experiments on four distinct types of spatio-temporal data tasks to evaluate the effectiveness of our proposed framework. The results demonstrate that our framework substantially improves the generalization capabilities in downstream prediction tasks across different spatio-temporal datasets. The noteworthy enhancement in predictive performance, achieved with model efficiency, underscores the efficacy of our FlashST framework.

2. Related Work

Deep Spatio-Temporal Learning is a thriving research field that aims to model and understand the intricate spatio-temporal dynamics in real-world urban data. Various approaches have been proposed, including integrating recurrent neural networks (RNNs) (Yao et al., 2018; Zhao et al., 2020) or Transformers(Wang et al., 2020; Xu et al., 2020) to capture long-term patterns and short-term fluctuations. Attention mechanisms have been widely used to dynamically weigh spatial and temporal features (Zheng et al., 2020; Zhang et al., 2021). MLPs are also utilized as temporal encoders to capture temporal correlations (Shao et al., 2022a). Another important direction involves leveraging Graph Neural Networks (GNNs) (Yu et al., 2018; Li et al., 2018; Yu et al., 2018; Ye et al., 2021) to simultaneously capture spatial dependencies and temporal evolutions. These models employ graph structures to represent relationships between spatial entities, facilitating information propagation. However, most existing deep spatio-temporal models adopt an end-to-end approach, limiting their generalization across diverse urban forecasting tasks and domains.

Pre-Training with Spatio-Temporal Data. There has been a recent surge of interest in using pre-training for acquiring comprehensive representations from spatio-temporal data through self-supervised learning. This approach aims to learn generalizable representations that capture the underlying patterns and dynamics. Two dominant research lines have emerged: i) integrating contrastive learning into the pre-training stage for spatio-temporal data (Zhang et al., 2023a), and ii) utilizing masked autoencoders (Shao et al., 2022b; Li et al., 2023) to reconstruct the masked input data to learn the spatio-temporal relationships.

Prompt-Tuning is a technique that optimizes prompts or instructions during inference to fine-tune the model, resulting in more accurate and context-specific predictions (Shrivastava et al., 2023; Liu et al., 2023a). It has gained recognition as a promising approach to improve the performance and adaptability of pre-trained models in various domains. For example, in prompt-based language generation, it allows precise control over the generated output, aligning it with specific requirements or desired styles (Brown et al., 2020; Schick & Schütze, 2020; Zhang et al., 2023b). Similarly, in vision prompt learning, specific instructions guide the model's attention towards relevant visual features or concepts (Zhou et al., 2022b;a; Sohn et al., 2023). This study explores spatio-temporal prompt-tuning to enhance prediction models in urban computing.

3. Preliminaries

Representation of Spatial-Temporal Data: The spatial-temporal information is captured and encoded using a three-

way tensor $\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{R \times T \times F}$. In this representation, R represents the number of regions, T denotes the time slots, and F indicates the number of features. Each entry $\mathbf{X}_{r,t,f}$ of the tensor corresponds to the f-th feature of the r-th region at the t-th time slot. For example, in the context of traffic flow prediction, the tensor $\mathbf{X}$ captures traffic volume data, which quantifies the number of vehicles passing through a specific region within a fixed time interval (e.g., 5 minutes).

Spatio-Temporal Prediction. In the context of spatio-temporal learning, a common scenario revolves around predicting future urban spatio-temporal conditions using historical records. This scenario can be described as follows:

$$\mathbf{X}^A_{t_{K+1}:t_{K+P}} = g(\mathbf{X}^A_{t_{K-H+1}:t_K}) \tag{1}$$

In this scenario, $\mathbf{X}^A$ represents the spatio-temporal data from dataset A. The function $g(\cdot)$ corresponds to the spatio-temporal predictive function, which could be methods like spatio-temporal transformers or spatio-temporal graph neural networks (GNNs). These predictive methods generate forecasts for the next P time slots by leveraging the historical spatio-temporal data from the previous H time slots.

$$\mathbf{X}^B_{t_{K+1}:t_{K+P}} = \tilde{g}(f(\mathbf{X}^B_{t_{K-H+1}:t_K})) \tag{2}$$

Furthermore, we introduce $\tilde{g}(\cdot)$ as the spatio-temporal prediction model with fixed parameters, and $\mathbf{X}^B$ represents an unseen spatio-temporal dataset. The goal of this work is to enable swift adaptation of the $\tilde{g}(\cdot)$ model to a new dataset while ensuring parameter efficiency. To achieve this, we propose incorporating a lightweight and effective prompt-tuning paradigm into the pre-trained $\tilde{g}(\cdot)$ model.

4. Methodology

This section provides an in-depth explanation of the technical details of the proposed FlashST framework. We commence by outlining the foundational paradigm and design principles that guide the development of the FlashST framework. Subsequently, we present a comprehensive overview of the framework's main components: the Spatio-Temporal In-Context Learning and the Unified Distribution Mapping Mechanism. To provide a visual representation of the overall model architecture, please refer to Figure 2.

4.1. Spatio-Temporal In-Context Learning

Our in-context learning framework is implemented through a spatio-temporal prompt network, which comprises two primary components: (i) Spatio-temporal context distillation mechanism efficiently captures contextual signals that are aware of both time and location from the unseen data. By doing so, it enables the model to learn from the specific context of the data, facilitating effective adaptation to various spatio-temporal scenarios. (ii) Spatio-temporal dependency modeling scheme incorporates complex relationships across time and locations into the in-context spatio-temporal network. By capturing and modeling these dependencies, the network can effectively understand the interdependencies and interactions between different spatio-temporal elements.

4.1.1. Spatio-Temporal Context Distillation

Spatio-Temporal Data Projection. To initialize the representation of the spatio-temporal data, we employ a projection layer with two steps: normalization using the Z-Score function and augmentation through linear transformation:

$$\mathbf{E}_{r,f} = \bar{\mathbf{X}}_{r,f}\mathbf{e}_0; \quad \bar{\mathbf{X}}_{r,t,f} = \mathrm{Z}(\mathbf{X}_{r,t,f}) = \frac{\mathbf{X}_{r,t,f} - \mu}{\sigma} \tag{3}$$

$\mathbf{E}_{r,f} \in \mathbb{R}^d$ represents the representation for the r-th region of the f-th feature, where d denotes the number of hidden units. $\bar{\mathbf{X}}_{r,t,f} \in \mathbb{R}$ corresponds to the normalized spatio-temporal data feature obtained through the Z-Score ($\mathrm{Z}(\cdot)$) function. The parameters μ and σ represent the mean and standard deviation of the raw ST matrix $\mathbf{X}$. The parametric vector $\mathbf{e}_0 \in \mathbb{R}^{T \times d}$ denotes the unit embedding vector for the normalized spatio-temporal feature.

Temporal Context Incorporation. In order to capture dynamic and periodic spatio-temporal patterns from diverse urban data, we introduce the time-aware context into our prompt network. This context is based on multi-resolution temporal features, specifically the hour of the day ($\mathbf{z}^{(d)} \in \mathbb{R}^T$) and the day of the week ($\mathbf{z}^{(w)} \in \mathbb{R}^T$). To effectively extract the temporal contextual signals, we employ a process of transformation and concatenation (denoted as $\mathrm{CAT}[,]$), which can be summarized as follows:

$$\mathbf{M}_t = \mathrm{CAT}[\mathbf{z}^{(d)}_t \cdot \mathbf{e}_1, \mathbf{z}^{(w)}_t \cdot \mathbf{e}_2] \tag{4}$$

Spatial Context Incorporation. To enhance the prompt network with geographical contextual information related to regional properties, we incorporate the citywide road network structure as an encoding feature that reflects spatial context. This process starts by formulating the normalized Laplacian matrix, which is defined as follows:

$$\triangle = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{5}$$

Here, $\mathbf{I}$, $\mathbf{D}$, and $\mathbf{A}$ denote the identity matrix, the degree matrix, and the adjacency matrix, respectively. The adjacency matrix is computed by considering the distances between areas and the road structure. Since Laplacian eigenvectors effectively preserve global graph structure information in Euclidean space, we perform eigenvalue decomposition to obtain $\triangle = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$. After extracting the eigenvalue matrix $\mathbf{\Lambda}$ and the corresponding eigenvector matrix $\mathbf{U}$, we derive the structure-aware node properties $\mathbf{C} \in \mathbb{R}^{R \times d_r}$ by projecting $\mathbf{U}$ to obtain the d_r smallest nontrivial eigenvectors.

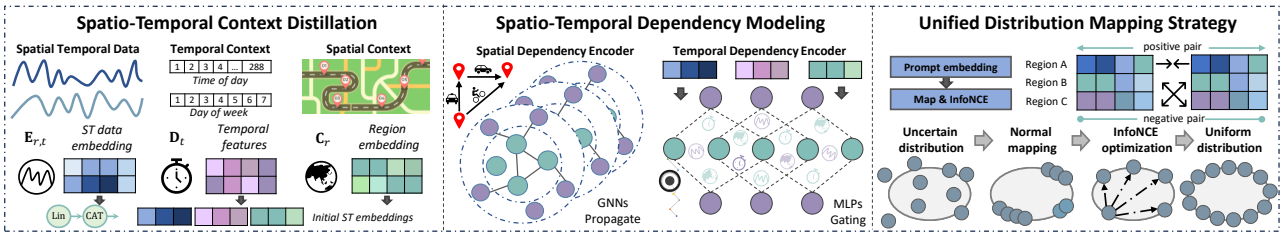


Figure 2. Our proposed FlashST framework adopts an architecture that integrates spatio-temporal in-context learning and a unified distribution mapping mechanism, offering an efficient and effective approach for spatio-temporal prompt-tuning across diverse scenarios.

Acknowledging the potential disparity of $\mathbf{C}$ in the feature space between training and test datasets, we employ a MLP to map these features and enhance the network's ability to generalize spatial context. Subsequently, we use the concatenation operation to integrate the aforementioned embeddings and obtain the initial spatio-temporal embedding:

$$\bar{\mathbf{E}} = \mathrm{CAT}[\mathbf{E}, \mathbf{M}, \bar{\mathbf{C}}]; \quad \bar{\mathbf{C}}_{r,f} = \mathrm{MLP}(\mathbf{C}_{r,f}) \tag{6}$$

$\bar{\mathbf{E}}$ denotes the integrated embeddings, where we concatenate the original representation $\mathbf{E}$, the temporal context embeddings $\mathbf{M}$, and the transformed regional properties $\bar{\mathbf{C}}$. The transformation of $\mathbf{C}$ is achieved through the MLP function.

4.1.2. Spatio-Temporal Dependency Modeling

Temporal Dependency Encoder. To capture dependencies across different time slots and preserve time-evolving data patterns, we introduce a lightweight gating mechanism. The formal operations of this mechanism are presented below:

$$\mathbf{H}_{r,f} = (W_2 \sigma(W_1 \bar{\mathbf{E}}_{r,f} + b_1) + b_2) + \bar{\mathbf{E}}_{r,f} \tag{7}$$

The trainable parameters of this gating mechanism are denoted by $W_1, W_2 \in \mathbb{R}^{d' \times d'}$ and $b_1, b_2 \in \mathbb{R}^{d'}$. The initial spatio-temporal embedding is represented by $\bar{\mathbf{E}}_{r,f}$, while the encoded temporal embedding is denoted by $\mathbf{H}_{r,f} \in \mathbb{R}^{d' \times d'}$. The embedding $\bar{\mathbf{E}}_{r,f}$ contains valuable contextual information about temporal dynamics and regional characteristics, which is essential for the temporal dependency encoder. This enriched information enables our in-context learning to effectively identify variations in spatio-temporal patterns across different regions and time intervals, thereby facilitating precise modeling of temporal correlations.

Spatial Dependency Encoder. Drawing inspiration from the effectiveness of graph neural networks in capturing spatial correlations between geographical locations (Wu et al., 2020b; Zheng et al., 2020), we leverage graph convolution-based message passing to encode inter-regional correlations. The adjacency matrix $\mathbf{A}$, as defined in Eq 5, acts as the connectivity matrix within the graph network framework. The process of spatial encoding is formalized as follows:

$$\mathbf{S}_f = \sigma(\mathbf{A} \cdot \mathbf{H}_f \cdot W_3) + \mathbf{H}_f \tag{8}$$

Here, $W_3 \in \mathbb{R}^{d' \times d'}$ represents the trainable parameters. To mitigate the potential over-smoothing phenomenon caused by multi-layer GNNs, we employ residual networks. By stacking multiple layers of spatio-temporal encoders, our model generates a representation $\mathbf{E}_{pro}$ that captures rich spatial and temporal dependencies across time and space.

4.2. Unified Distribution Mapping Mechanism

To bridge the distribution gap between the pre-training and diverse unseen data in downstream tasks, we enhance FlashST by incorporating a distribution mapping mechanism. The objective of this mechanism is to transform both the pre-training data and the downstream data into a shared distribution space. This alignment of data distributions enables a seamless transfer of knowledge, ensuring that the insights gained from the pre-training stage can be effectively applied to the downstream spatio-temporal context.

To achieve this goal, our FlashST involves standardizing the prompt embedding to ensure a consistent distribution across diverse downstream datasets. We draw inspiration from various works in contrastive learning (Wang & Isola, 2020; Chuang et al., 2020; Wang & Liu, 2021) and incorporate the infoNCE-based loss function to regularize the prompt network for representation generation. This loss function is designed to bring the representations of positive sample pairs closer together while pushing apart those of negative pairs. By leveraging self-supervised learning without the need for additional labeling, optimizing the infoNCE loss helps attain a more uniform embedding distribution. Empirical evidence suggests that through this loss alone, an almost entirely uniform distribution can be achieved (Yu et al., 2022). Building upon this, we employ the infoNCE loss to adjust the distribution of the learned spatio-temporal prompt embedding $\mathbf{E}_{pro} \in \mathbb{R}^{R \times F \times d'}$, formalized as:

$$\mathcal{L}_{Uni} = \sum_{r=1}^{R} \sum_{f=1}^{F} -\log \sum_{r'} \exp(\cos(\mathbf{E}_{pro}^{r,f}, \mathbf{E}_{pro}^{r',f})/\tau) \tag{9}$$

The cosine similarity function $\cos(\cdot)$ is used to measure the similarity between embeddings, and the temperature coefficient τ is used to adjust the softmax scale. In our FlashST, we enhance the uniformity of prompt embeddings by increasing the separation among embeddings that correspond

to different regions. This improvement allows downstream models to effectively utilize the provided prompts $\mathbf{E}_{pro}^{(U)}$ for rapid generalization across new data and tasks.

4.3. Unifying Pre-training and Downstream Tasks

Our spatio-temporal pretrain-prompt framework integrates the model pre-training process with downstream forecasting tasks through efficient prompt-tuning. In the pre-training phase, we train and optimize all parameters using dedicated pre-training datasets. Subsequently, in the prompt fine-tuning phase, we exclusively update the parameters of the prompt network by conducting a limited number of training epochs on unseen datasets. This approach enables the downstream model to effectively adapt to new data. Moreover, our framework is model-agnostic, allowing seamless integration with various existing spatio-temporal prediction baselines as the downstream model.

i) **Pre-training Phase**: Our goal is to forecast future trends using historical spatio-temporal records from the pre-training data. We achieve this with following process:

$$\hat{\mathbf{Y}}^{A} = g(\mathbf{E}_{pro}^{(U)}); \quad \mathbf{E}_{pro}^{(U)} = f_{Prompt}(\mathbf{X}^{A}) \tag{10}$$

Here, we have adjustable parameters for both $f_{Prompt}(\cdot)$ and $g(\cdot)$. To update these parameters, we utilize the loss function $\mathcal{L}$ in conjunction with the Adam optimizer. The loss function $\mathcal{L}$ is a combination of the regression loss $\mathcal{L}_r$ and the uniformity loss $\mathcal{L}_{Uni}$, with λ as the balance coefficient between the two losses. The label is represented by $Y \in \mathbb{R}^{R\times T\times F}$. The regression loss $\mathcal{L}_r$ is defined as:

$$\mathcal{L}_r = \frac{1}{RTF}\sum_{r=1}^{R}\sum_{t=1}^{T}\sum_{f=1}^{F}|(\mathbf{Y}_{r,t,f} - \hat{\mathbf{Y}}_{r,t,f})| \tag{11}$$

This formulation calculates the average absolute difference between the predicted values $\hat{\mathbf{Y}}_{r,t,f}$ and the actual labels $\mathbf{Y}_{r,t,f}$ across all regions R, time steps T, and features F.

ii) **Prompt-Tuning Phase**: We freeze the parameters of the downstream model and focus exclusively on fine-tuning the prompt network. This fine-tuning process is carried out on the test dataset to perform spatio-temporal prediction tasks, following the formulation described in Eq 10:

$$\hat{\mathbf{Y}}^{B} = \tilde{g}(f_{Prompt}(\mathbf{X}^{B})) \tag{12}$$

Here, $\tilde{g}$ represents the downstream model with frozen parameters, and B indicates the test datasets. During the prompt-tuning phase, we conduct a limited number of optimization epochs to generate high-quality spatio-temporal prompt representations for the downstream models.

5. Evaluation

In this section, we analyze the performance of our model by addressing the following several research questions:

Table 1. Statistics of pre-training dataset.

Dataset	Data Record	# Region	Time Steps	Sample Date
PEMS03	traffic flow	358	26208	1/Sep/2018 - 30/Nov/2018
PEMS04	traffic flow	307	16992	1/Jan/2018 - 28/Feb/2018
PEMS07	traffic flow	883	26208	1/May/2017 - 30/Aug/2017
PEMS08	traffic flow	170	17856	1/Jul/2016 - 31/Aug/2016

Table 2. Statistics of downstream tasks dataset.

Dataset	Data Record	# Region	Time Steps	Sample Date
PEMS07(M)	traffic speed	170	12672	1/Jul/2016 - 31/Aug/2016
CA-D5	traffic flow	211	16992	1/Jan/2017 - 28/Feb/2017
ChengDu-DIDI	traffic index	524	17280	1/Jan/2018 - 30/Apr/2018
NYC Citi Bike	bike orders	250	4368	1/Apr/2016 - 30/Jun/2016

- **RQ1**: Does FlashST effectively generalize pre-trained models to new spatio-temporal prediction data and tasks?
- **RQ2**: How does the efficiency of our model compare to that of end-to-end training and fine-tuning approaches?
- **RQ3**: What is the impact of the key components of our FlashST on the performance of downstream models?
- **RQ4**: How do parameters affect the model performance?

5.1. Experimental Setting

Datasets. To evaluate how well the model can generalize across different urban spatio-temporal contexts, our experimental setup is designed as follows: (i) In the pre-training phase, we use four datasets (PEMS03, PEMS04, PEMS07, and PEMS08 (Song et al., 2020)) as our training sets. These datasets consist of records detailing traffic flow conditions across various streets and cities in California, USA. (ii) During the subsequent prompt-tuning phase, we concentrate on four distinct target datasets to fine-tune and evaluate our framework: PEMS07(M) (Yu et al., 2018), CA-D5 (Liu et al., 2023b), ChengDu-DIDI (Lu et al., 2022), and NYC Citi Bike (Ye et al., 2021). These datasets correspondingly represent traffic speeds in Los Angeles, traffic flow in California, traffic flow in Chengdu, and bicycle demand in New York City. Each target dataset is partitioned into training, validation, and test sets in a ratio of 6:2:2. Detailed statistics for each dataset can be found in Table 1 and Table 2.

Experimental Description. We configure the model with 32 hidden units for d, d_t, and d_r, while employing a spatial and temporal encoder with 2 layers. The pre-training phase involves alternating training for 300 epochs on the four pre-training datasets. Subsequently, we conduct fine-tuning for 20 epochs during the prompt-tuning phase. In both the baseline and full fine-tuning setups, the maximum number of training epochs is limited to 100. Furthermore, a batch size of 64 is employed for both phases and all baselines. For the dataset-specific parameters in the baselines (*e.g.*, node embeddings), we initialize them randomly and make them trainable during the prompt fine-tuning phase. Additionally, the final regression layers of some baseline models are set to be trainable to ensure accurate numerical predictions.

Evaluation Metrics. In order to assess the performance of the model in spatiotemporal prediction, we employ three widely-used evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean

Table 3. Overall performance on PEMS07(M), CA-D5, ChengDu-DIDI and NYC Citi Bike datasets in terms of *MAE*, *RMSE* and *MAPE*.

Model	Dataset	PEMS07(M)			CA-D5			ChengDu-DIDI			NYC Citi Bike		
	Metrics	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
TGCN		5.05	8.56	13.68%	15.36	24.52	24.42%	3.18	4.72	14.48%	2.19	3.54	63.31%
STGCN		3.14	6.25	8.05%	15.02	23.73	27.84%	2.59	3.97	12.20%	2.04	3.21	52.06%
ASTGCN		3.13	6.19	8.09%	15.26	24.37	24.45%	2.56	3.83	11.58%	2.03	3.10	55.13%
GWN		2.67	5.40	6.62%	13.63	22.30	24.68%	2.35	3.56	10.72%	1.81	2.71	50.05%
STSGCN		2.93	5.87	7.52%	15.30	24.44	24.70%	2.86	4.29	13.02%	1.89	2.78	52.48%
AGCRN		2.89	5.69	7.32%	14.44	23.10	23.02%	2.87	4.27	13.05%	2.16	3.41	60.95%
MTGNN		2.70	5.50	6.81%	13.90	22.44	23.86%	2.33	3.56	10.63%	1.84	2.80	**48.69%**
STFGNN		3.14	6.21	7.81%	15.38	24.37	25.78%	2.83	4.24	13.05%	2.36	3.73	61.93%
STGODE		2.81	5.57	7.09%	14.13	22.77	23.41%	2.51	3.75	11.48%	1.94	2.83	56.25%
DMSTGCN		2.94	5.92	7.45%	13.88	22.81	21.97%	2.67	4.00	12.34%	2.09	3.32	60.00%
MSDR		2.79	5.53	7.07%	15.31	23.71	33.01%	2.56	3.85	11.94%	1.87	2.86	51.25%
STWA		2.74	5.52	6.97%	14.00	22.75	23.22%	2.44	3.68	11.33%	1.88	2.83	49.93%
PDFormer		2.68	5.46	6.66%	13.85	22.59	23.11%	2.43	3.70	11.17%	1.89	2.88	51.98%
Ours		**2.59**	**5.25**	**6.54%**	**13.26**	**21.83**	**21.49%**	**2.31**	**3.52**	**10.51%**	**1.79**	**2.63**	49.77%

Absolute Percentage Error (MAPE). These metrics quantify the disparities between the predicted data and the ground truth data. It is important to note that lower values of these metrics indicate better performance.

Baseline Description. To evaluate the effectiveness of our FlashST, we selected 13 advanced spatio-temporal prediction models as baselines. These include RNNs-based models, attention-based models, GNNs-based models, and differential equation-based spatio-temporal prediction methods.

RNN-based Spatio-Temporal Prediction Approaches:

- **AGCRN** (Bai et al., 2020): This method incorporates RNNs with learnable node embeddings to capture personalized spatio-temporal patterns of regions.
- **MSDR** (Liu et al., 2022): This model addresses the issue of long-term information forgetting in RNNs by introducing a variant of RNNs. It further combines this variant with GNNs to capture spatio-temporal correlations.

GNN-based Spatio-Temporal Prediction Models:

- **TGCN** (Zhao et al., 2020): This model leverages a combination of RNNs and GCNs to separately model temporal dependencies and spatial correlations.
- **STGCN** (Yu et al., 2018): This method utilizes gated convolutional networks to encode temporal dependencies and GCNs to capture local spatial relationships.
- **GWN** (Wu et al., 2019): This approach incorporates a learnable graph structure to model spatial correlations, while TCNs are used to extract temporal features.
- **STSGCN** (Song et al., 2020): It synchronously models dynamic spatio-temporal correlations by constructing a graph network that allows interactions across time slices.
- **MTGNN** (Wu et al., 2020b): This method captures temporal dependencies using time convolutional networks combined with skip connections, and employs learnable graph networks to model spatial correlations.
- **STFGNN** (Li & Zhu, 2021): This method proposes a data-driven spatio-temporal graph construction approach to capture dynamic spatio-temporal correlations.
- **DMSTGCN** (Han et al., 2021): This method models dynamic spatial correlations by constructing a learnable dynamic graph network.

Attention-based Spatio-Temporal Models:

- **ASTGCN** (Guo et al., 2019): This method leverages attention mechanisms to effectively capture the periodic spatio-temporal correlations inherent in the data.
- **STWA**(Cirstea et al., 2022): This approach goes beyond standard attention mechanisms by incorporating specialized node features and time-dynamic parameters.
- **PDFormer** (Jiang et al., 2023): This model leverages the power of the transformer architecture to effectively encode the intricate spatio-temporal dependencies present in the data. Furthermore, it introduces a novel time-delayed spatial correlation modeling technique.

ODE-based Spatio-Temporal Prediction Methods:

- **STGODE** (Fang et al., 2021): This method enriches graph networks with ordinary differential equations (ODEs) to extract continuous spatial correlation information. Time convolutional networks are employed to learn time-dependent relationships within the data.

Table 4. Model-agnostic experiments on PEMS07(M), CA-D5, ChengDu-DIDI and NYC Citi Bike datasets.

Model	Dataset	PEMS07(M)			CA-D5			ChengDu-DIDI			NYC Citi Bike		
	Metrics	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN		3.14	6.25	8.05%	15.02	23.73	27.84%	2.59	3.97	12.20%	2.04	3.21	52.06%
w/o Finetune		8.07	11.77	23.02%	31.20	41.82	68.75%	4.82	6.53	22.87%	4.10	5.60	148.20%
w/ Finetune		3.18	6.25	8.04%	15.15	24.18	23.30%	2.58	3.93	12.01%	1.99	3.11	50.38%
Ours		**2.68**	**5.37**	**6.80%**	**13.47**	**22.01**	**22.62%**	**2.37**	**3.60**	**10.92%**	**1.80**	**2.67**	**50.06%**
GWN		2.67	5.40	**6.62%**	13.63	22.30	24.68%	2.35	3.56	10.72%	1.81	2.71	**50.05%**
w/o Finetune		7.41	12.78	24.30%	27.28	41.46	48.26%	4.89	6.99	23.42%	3.39	4.99	112.57%
w/ Finetune		2.69	5.38	6.74%	13.61	22.44	22.82%	**2.35**	**3.55**	**10.72%**	1.85	2.79	51.88%
Ours		**2.67**	**5.36**	6.79%	**13.19**	**21.68**	**21.21%**	2.37	3.58	10.77%	**1.80**	**2.66**	51.16%
MTGNN		2.70	5.50	6.81%	13.90	22.44	23.86%	2.33	3.56	10.63%	1.84	2.80	**48.69%**
w/o Finetune		5.91	9.21	14.56%	50.07	65.35	77.26%	4.45	6.27	20.06%	3.32	4.74	118.01%
w/ Finetune		2.62	5.33	6.56%	13.46	22.02	22.10%	2.33	3.54	10.52%	1.81	2.74	49.35%
Ours		**2.59**	**5.25**	**6.54%**	**13.26**	**21.83**	**21.49%**	**2.31**	**3.52**	**10.51%**	**1.79**	**2.63**	49.77%
PDFormer		2.68	5.46	**6.66%**	13.85	22.59	23.11%	2.43	3.70	11.17%	1.89	2.88	51.98%
w/o Finetune		5.57	10.03	19.16%	20.45	31.56	30.21%	3.60	5.20	16.93%	3.19	4.76	111.40%
w/ Finetune		2.71	5.44	6.80%	13.77	22.31	24.90%	2.43	3.69	11.15%	1.83	2.78	**48.56%**
Ours		**2.68**	**5.31**	6.79%	**13.55**	**22.08**	**21.98%**	**2.41**	**3.64**	**11.05%**	**1.81**	**2.68**	52.06%

5.2. Overall Performance (RQ1)

In this section, we investigate the effectiveness of FlashST in enhancing the generalization capabilities of downstream models across new spatio-temporal contexts. As shown in Table 3, the baseline models were exclusively trained and tested on the target dataset to evaluate their performance.

Superiority over End-to-End Prediction Models. Upon analyzing the results presented in Table 3, it becomes evident that our approach exhibits a significant advantage over the end-to-end spatio-temporal models in diverse urban data prediction scenarios. These findings provide compelling evidence of the effectiveness of our FlashST method in accurately capturing the intricate spatio-temporal invariant patterns present in urban data. Our new in-context learning paradigm excels in transferring this acquired knowledge to adapt the pre-trained model to new downstream tasks. By effectively handling the distribution gap, our method bridges the semantic gap between the pre-trained model and the specific prediction scenarios encountered in practice.

Model-Agnostic Advantage. The model-agnostic advantage is a notable strength of our approach. We have designed our method to seamlessly integrate with various spatio-temporal backbone encoders, providing flexibility and avoiding the constraints of a specific model choice. This adaptability is demonstrated in Table 4, where our FlashST approach showcases easy adaptation with four state-of-the-art spatio-temporal models (i.e., STGCN, GWN, MTGNN, PDFormer). The evaluation results highlight the versatility of our FlashST approach, showcasing its exceptional performance enhancement when combined with these cutting-edge spatio-temporal models. The successful integration of our approach with state-of-the-art models further reinforces its adaptability and capability to improve prediction accuracy in diverse urban data scenarios.

Comparison with Model Fine-Tuning. To further demonstrate the effectiveness of our spatio-temporal in-context framework, we compare our prompt-tuning approach with the full fine-tuning of the model parameters. The "w/o Finetune" approach refers to direct prediction on the target dataset without any fine-tuning after pre-training. On the other hand, "w/ Finetune" indicates the utilization of full-parameter fine-tuning to adapt the models to the target data after pre-training. However, it is worth noting that the observed improvements with direct fine-tuning, compared to end-to-end prediction outcomes, suggest that pre-training may not have provided a significant advantage. In the absence of effective alignment between the pre-trained model and downstream tasks, noise can be introduced, leading to misleading fine-tuning and suboptimal performance.

5.3. Assessment of Model Efficiency (RQ2)

Training Time. In this section, we focus on evaluating the efficiency of our proposed method. We measured the training time for three different scenarios: end-to-end training, full fine-tuning, and our FlashST approach. The results are presented in Table 5. For end-to-end training and full fine-tuning, we followed the settings of existing baselines, configuring the training epochs to 100. We also set an early stopping criterion of 25, which halts the training process if no decrease in validation loss is observed. Regarding our FlashST approach, we limited the number of epochs for prompt-tuning to 20. This constraint was implemented to promote swift adaptation of the downstream models to the new datasets. The results suggest that the same baseline model showcases comparable efficiency in both end-to-end training and full-parameter fine-tuning. The disparity in training time between these two settings primarily arises from variations in convergence speed caused by different initialization parameters. One of the significant contribu-

Table 5. Computational time cost investigation (seconds).

Model	time(s)	Model	time(s)
ASTGCN	1310	TGCN	927
AGCRN	689	STSGCN	2253
DMSTGCN	1330	STFGNN	990
MSDR	5190	STWA	5143
STGCN	411	GWN	1042
STGCN Finetune	445	GWN Finetune	1185
STGCN w/ Ours	190	GWN w/ Ours	222
MTGNN	962	PDFormer	7524
MTGNN Finetune	1095	PDFormer Finetune	6466
MTGNN w/ Ours	230	PDFormer w/ Ours	1220

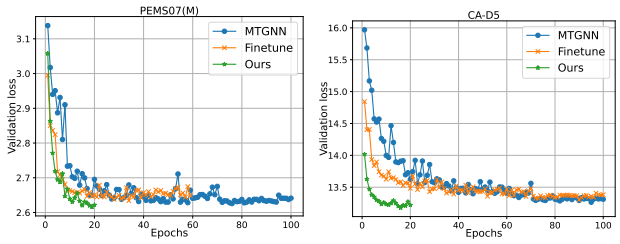


Figure 3. The convergence efficiency of FlashST.

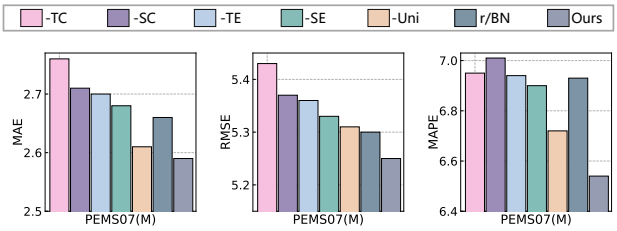
tions of our FlashST framework is the notable increase in computational efficiency. It reduces the training time of baseline models by 20% to 80%. This substantial enhancement significantly improves their efficiency in adapting to new spatio-temporal data.

Faster Convergence Rate. To further validate the ability of our FlashST framework to swiftly adapt to new spatio-temporal scenarios, we conducted an investigation into its convergence speed on different datasets. We specifically focused on the decreasing trend of the validation error when using the PEMS07(M) and CA-D5 datasets, with MTGNN adopted as the downstream model. The results are visualized in Figure 3. The findings clearly demonstrate that with the integration of our FlashST approach, the downstream model achieves convergence within a few tuning epochs. In contrast, the end-to-end training and fine-tuning paradigms require a greater number of training rounds to fit the new data. This phenomenon can be attributed to the effectiveness of our proposed spatio-temporal prompt network and data mapping strategy. These components enable the model to leverage the spatio-temporal characteristics of the new data in conjunction with the pre-trained knowledge, facilitating rapid adaptation to diverse spatio-temporal scenes.

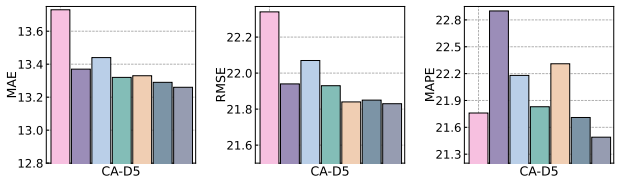
5.4. Ablation Study (RQ3)

In this section, we conduct an ablation study to investigate the contributions of each key module in our FlashST framework. The corresponding experimental results are depicted in Figure 4. We utilize MTGNN as the downstream baseline and assess the performance of each component on the PEMS07(M) and CA-D5 data. Before the prompt-tuning step, we conduct re-pre-training to adapt to each variant.

- **Impact of Spatio-Temporal Context Distillation.** To assess the importance of temporal and spatial context distillation, we conducted experiments by removing the distillation with temporal context (-TC) and spatial context (-SC) separately. The results clearly demonstrate a substantial performance drop across most metrics when the distilled temporal and spatial context are removed. This highlights the critical importance of preserving both temporal and spatial context during the in-context learning process. Effective encoding of the temporal information and the incorporation of the spatial information are vital for capturing dynamic periodic temporal patterns and region-specific properties, thereby enhancing the model's ability to recognize spatio-temporal invariant patterns and improve its data understanding.
- **Impact of Spatio-Temporal Dependency Modeling.** In our analysis, we individually removed the temporal encoder (-TE) and spatial encoder (-SE) to investigate their contributions. Our analysis revealed that spatio-temporal dependency encoding plays a vital role in effectively integrating complex relationships among different time slots and locations during the in-context learning process. The inclusion of both temporal and spatial dependency encoders enables the model to comprehend and leverage the intricate interactions between time and space. This proficiency facilitates a swifter adaptation of downstream models to new spatio-temporal scenarios.
- **Impact of Unified Distribution Mapping Mechanism.** In our variants, we assess the utility of the unified distribution mapping strategy through two aspects. **i) -Uni.** We did not explore the unified distribution mapping strategy. However, the decline in performance indicates its positive impact on the model. By mapping the diverse spatio-temporal data embeddings into a uniform distribution, our FlashST effectively mitigates the influence of distributional disparities between the pre-training data and unseen spatio-temporal data. **ii) r/BN.** The unified distribution mapping strategy has been replaced with batch normalization. Batch normalization standardizes the data based on local statistical properties of mini-batches, mitigating the issue of internal covariate shift during neural network training and thereby enhancing convergence efficiency. However, due to the lack of an established connection between the pretrained data and the downstream task data, it is challenging for the downstream model to effectively transfer knowledge from the pretraining process. The unified distribution mapping strategy ensures that the model can effectively leverage the knowledge acquired during the pre-training. By aligning the distributions of different data sources, the model can better adapt to new spatio-temporal scenarios and make more accurate predictions.



(b) MAE, RMSE and MAPE on PEMS07(M) dataset.



(c) MAE, RMSE and MAPE on CA-D5 dataset.

Figure 4. Ablation study of FlashST.

5.5. Hyperparameter Analysis (RQ4)

In this section, we investigate the impact of different hyperparameter settings, specifically across various temperature coefficients and loss weight coefficients, on the model's performance. Our findings reveal that the model achieves optimal performance with a parameter configuration of $\tau = 0.3$ and $\lambda = 1.0$, as shown in Figure 6. Notably, variations in these parameters minimally influence the final result, highlighting the model's effective adaptability to different parameters settings. It learns efficient representations that distinguish embedded features in various regions, even with differences in feature scales. Additionally, the model's performance remains unaffected by an increase in the uniformity loss. This indicates that the key component of our distribution mapping strategy does not interfere with the predictive loss. This further supports the viability of our strategy and facilitates the downstream model's rapid generalization to novel spatio-temporal contexts.

5.6. Case Study (RQ5)

To assess the effectiveness of our proposed uniform distribution mapping approach in transforming various data representations into uniform distributions, we conducted visualizations of prompt embeddings with and without the application of our distribution mapping mechanism. These visualizations were performed using two variants of the downstream model MTGNN and evaluated on the test sets of the PEMS07(M) and CA-D5 datasets. Given that prompt embeddings are high-dimensional vectors, we initially applied the PCA technique to reduce the dimensionality of each embedding sample to 2-D (Wold et al., 1987). Subsequently, we projected the reduced embeddings onto the unit circle using the L2 norm, as depicted in Figure 6. The visualization results provide compelling evidence that our uniform distribution mapping strategy effectively transforms the prompt embeddings into an approximate uniform distribution. In contrast, the variants that lack this strategy fail to achieve such desirable distribution properties. By converting data from new spatio-temporal contexts into a consistent distribution, our FlashST gains the ability to utilize pre-trained knowledge and rapidly adapt to new datasets, facilitating its performance on various spatio-temporal tasks.

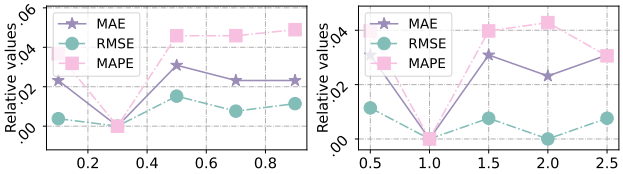


Figure 5. Hyperparameter study of τ and λ.

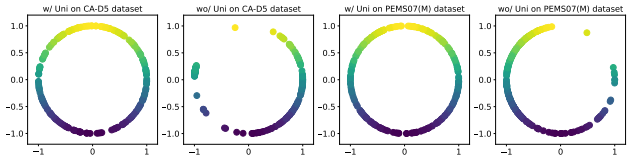


Figure 6. Distribution of the prompt embedding.

6. Conclusion

This paper presents FlashST, a new and efficient method for adapting spatio-temporal predictive models to various downstream tasks that involve previously unseen data. Our in-context learning framework utilizes a spatio-temporal prompt network, which consists of a context distillation mechanism and a dependency modeling scheme. This framework effectively adapts to different spatio-temporal scenarios by capturing contextual signals and modeling complex relationships across time and locations. To address the distribution gap, we enhance FlashST by incorporating a distribution mapping mechanism that aligns the data distributions of pre-training and downstream data, facilitating effective knowledge transfer in spatio-temporal forecasting. Extensive experiments demonstrate the effectiveness and generalization capabilities of our FlashST in diverse downstream spatio-temporal forecasting scenarios. For future work, an interesting direction would be to explore the potential of incorporating large language models as knowledge guidance within our FlashST framework.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

This work is partially supported by the Guangzhou Municipal Science and Technology Plan Project - Key Research and Development Program (Grant No. 2024B01W0007).

References

Bai, L., Yao, L., Li, C., Wang, X., and Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. In *NeurIPS*, pp. 17804–17815, 2020.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. Language models are few-shot learners. In *NeurIPS*, pp. 1877–1901, 2020.

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. Debiased contrastive learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *NeurIPS*, pp. 8765–8775, 2020.

Cirstea, R.-G., Yang, B., Guo, C., Kieu, T., and Pan, S. Towards spatio- temporal aware traffic time series forecasting. In *ICDE*, pp. 2900–2913, 2022.

Fang, Z., Long, Q., Song, G., and Xie, K. Spatial-temporal graph ode networks for traffic flow forecasting. In *KDD*, pp. 364–373, 2021.

Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*, pp. 922–929, 2019.

Han, L., Du, B., Sun, L., Fu, Y., Lv, Y., and Xiong, H. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *KDD*, pp. 547–555, 2021.

Jiang, J., Han, C., Zhao, W. X., and Wang, J. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *AAAI*, pp. 4365–4373, 2023.

Li, M. and Zhu, Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *AAAI*, pp. 4189–4196, 2021.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *ICLR*, 2018.

Li, Z., Huang, C., Xia, L., Xu, Y., and Pei, J. Spatial-temporal hypergraph self-supervised learning for crime prediction. In *ICDE*, pp. 2984–2996, 2022.

Li, Z., Xia, L., Xu, Y., and Huang, C. Gpt-st: Generative pre-training of spatio-temporal graph neural networks. In *NeurIPS*, pp. 70229–70246, 2023.

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.

Liu, D., Wang, J., Shang, S., and Han, P. Msdr: Multi-step dependency relation networks for spatial temporal forecasting. In *KDD*, pp. 1042–1050, 2022.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, pp. 1–35, 2023a.

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*, 2023b.

Lu, B., Gan, X., Zhang, W., Yao, H., Fu, L., and Wang, X. Spatio-temporal graph few-shot learning with cross-city knowledge transfer. In *KDD*, pp. 1162–1172, 2022.

Schick, T. and Schütze, H. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020.

Shao, Z., Zhang, Z., Wang, F., Wei, W., and Xu, Y. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *CIKM*, pp. 4454–4458, 2022a.

Shao, Z., Zhang, Z., Wang, F., and Xu, Y. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *KDD*, pp. 1567–1577, 2022b.

Shrivastava, D., Larochelle, H., and Tarlow, D. Repository-level prompt generation for large language models of code. In *ICML*, pp. 31693–31715, 2023.

Sohn, K., Chang, H., Lezama, J., Polania, L., Zhang, H., Hao, Y., Essa, I., and Jiang, L. Visual prompt tuning for generative transfer learning. In *CVPR*, pp. 19840–19851, 2023.

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 *AAAI*, pp. 914–921, 2020.

Wang, F. and Liu, H. Understanding the behaviour of contrastive loss. In *CVPR*, pp. 2495–2504, June 2021.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pp. 9929–9939, 2020.

Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., and Yu, J. Traffic flow prediction via spatial temporal graph neural network. In *WWW*, pp. 1082–1092, 2020.

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, pp. 37–52, 1987.

Wu, T., Ren, H., Li, P., and Leskovec, J. Graph information bottleneck. In *NeurIPS*, pp. 20437–20448, 2020a.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*, pp. 1907–1913, 2019.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *KDD*, pp. 753–763, 2020b.

Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., and Xiong, H. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Chuxing, D., and Li, Z. Deep multi-view spatial-temporal network for taxi demand prediction. In *AAAI*, pp. 2588–2595, 2018.

Ye, J., Sun, L., Du, B., Fu, Y., and Xiong, H. Coupled layer-wise graph convolution for transportation demand prediction. In *AAAI*, pp. 4617–4625, 2021.

Yu, B., Yin, H., et al. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*, pp. 3634–3640, 2018.

Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., and Nguyen, Q. V. H. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *SIGIR*, pp. 1294–1303, 2022.

Yu, R., Li, Y., Shahabi, C., Demiryurek, U., and Liu, Y. Deep learning: A generic approach for extreme condition traffic forecasting. In *SDM*, pp. 777–785, 2017.

Zhang, J., Zheng, Y., and Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pp. 1655–1661, 2017.

Zhang, Q., Huang, C., Xia, L., Wang, Z., Yiu, S. M., and Han, R. Spatial-temporal graph learning with adversarial contrastive adaptation. In *ICML*, pp. 41151–41163, 2023a.

Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., and Qiao, Y. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023b.

Zhang, X., Huang, C., Xu, Y., Xia, L., Dai, P., Bo, L., Zhang, J., and Zheng, Y. Traffic flow forecasting with spatial-temporal graph diffusion network. In *AAAI*, pp. 15008–15015, 2021.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *Transactions on Transportation Systems (TITS)*, pp. 3848–3858, 2020.

Zheng, C., Fan, X., Wang, C., and Qi, J. Gman: A graph multi-attention network for traffic prediction. In *AAAI*, pp. 1234–1241, 2020.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Conditional prompt learning for vision-language models. In *CVPR*, pp. 16816–16825, 2022a.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, pp. 2337–2348, 2022b.