A Universal Pre-training and Prompting Framework for General Urban Spatio-Temporal Prediction

Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, *Member, IEEE* and Yong Li *Senior Member, IEEE*

***Abstract*—Urban spatio-temporal prediction is crucial for informed decision-making, such as traffic management, resource optimization, and emergence response. Despite remarkable breakthroughs in pretrained natural language models that enable one model to handle diverse tasks, a universal solution for spatio-temporal prediction remains challenging. Existing prediction approaches are typically tailored for specific spatio-temporal scenarios, requiring task-specific model designs and extensive domain-specific training data. In this study, we introduce UniST, a universal model designed for general urban spatio-temporal prediction across a wide range of scenarios. Inspired by large language models, UniST achieves success through: (i) utilizing diverse spatio-temporal data from different scenarios, (ii) effective pre-training to capture complex spatio-temporal dynamics, (iii) knowledge-guided prompts to enhance generalization capabilities. These designs together unlock the potential of building a universal model for various scenarios. Extensive experiments on more than 20 spatio-temporal scenarios, including grid-based data and graph-based data, demonstrate UniST's efficacy in advancing state-of-the-art performance, especially in few-shot and zero-shot prediction. The datasets and code implementation are released on https://github.com/tsinghua-fib-lab/UniST.**

***Index Terms*—Spatio-temporal prediction, prompt learning, universal model.**

I. Introduction

Pre-trained foundation models have showcased remarkable success in Natural Language Processing (NLP) [1], [2], particularly excelling in few-shot and zero-shot settings [2], [3]. However, similar breakthroughs have not yet been achieved in the field of urban spatio-temporal prediction [4]–[6]. Data scarcity is a critical challenge [7]–[9], as acquiring extensive datasets across diverse urban contexts is often infeasible due to privacy concerns, cost, and time constraints. This scarcity restricts the ability of traditional models to generalize across varied environments and impedes their performance in cities or scenarios with limited data. In this paper, our goal is to establish a foundation model for general urban spatio-temporal prediction — specifically, to develop a universal model that offers superior performance and powerful generalization capabilities across diverse spatio-temporal scenarios. This entails training a single model capable of effectively handling various urban contexts, encompassing various domains such as human mobility, traffic and communication networks across different cities.

The significance of such a universal model lies in its ability to address prevalent data scarcity issues in urban areas.

Y. Yuan, J. Ding, J. Feng, D. Jin, and Y. Li are with Beijing National Research Center for Information Science and Technology (BNRist), and Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

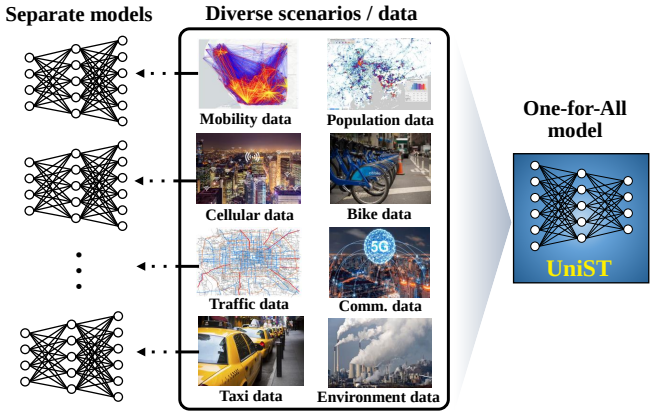


Fig. 1: The transition from traditional separate deep learning models to a one-for-all universal model for urban spatio-temporal prediction.

The varying levels of digitalization across domains and cities often result in imbalanced and incomplete datasets. Despite notable advancements in existing spatio-temporal modeling approaches [10]–[16], their effectiveness is typically confined to specific domains within a single city. The reliance on extensive training data further impedes the model's generalization potential. Consequently, current solutions are still far from "universality", and remain narrowly applicable.

A universal spatio-temporal model must possess two essential capabilities. *Firstly, it must be capable of leveraging abundant and rich data from different urban scenarios for training.* The training of the foundational model should ensure the acquisition of ample and rich information [1], [17], [18]. *Second, it should demonstrate robust generalization across different spatio-temporal scenarios*. Especially in scenarios with limited or no training data, the model can still work well without obvious performance degradation [17], [19].

However, realizing the aforementioned capabilities encounters significant challenges specific to spatio-temporal data, which impede the direct application of current foundation models developed for language and vision domains. The first challenge arises from the inherent *diverse formats* of spatio-temporal datasets. Unlike languages with a natural and unified sequential structure or images and videos adhering to standardized dimensions, spatio-temporal data collected from different sources exhibit highly varied features. These include variable dimensions, temporal durations, and spatial coverages that differ significantly, posing difficulties in standardizing their structure. Additionally, spatio-temporal data are organized in

distinct structures, such as grid-based spatial partitions and graph-based spatial correlations. The conference version [20] was limited to handling only grid-based data. The second challenge arises from *high variations in data distributions across multiple scenarios*. Faced with highly distinct spatio-temporal patterns, the model may struggle to adapt to these differences. Unlike language, which benefits from a shared vocabulary, various scenarios of different domains and cities often operate on entirely different spatial and temporal scales, lacking common elements for effective training and generalization.

Although the displayed spatio-temporal patterns vary significantly, there are certain underlying laws that should be common among them. This principle arises from the intuition that human activity influences various spatio-temporal data generated in urban settings, leading to the existence of universal patterns. For example, traffic speed and communication networks exhibit distinct spatio-temporal patterns, yet both are influenced by human mobility and therefore adhere to similar underlying principles. Additionally, while temporal periodic patterns vary across domains, they share fundamental concept of repetition. Furthermore, city layouts vary considerably between different urban areas, but the relationships among various functional zones within cities may exhibit shared characteristics. Therefore, the key to building a one-for-all model is to capture, align and leverage these shared underlying characteristics effectively, while accommodating both grid and graph data structures.

To this end, we introduce ***UniST***, a **uni**versal solution for urban **s**patio-**t**emporal prediction through advanced pre-training and prompt learning. Notably, UniST achieves three essential capabilities of:

1) Scalability across scenarios with diverse spatio-temporal data, including various domains, cities, and data structures;
2) Effective pre-training to capture complex spatio-temporal relationships;
3) utilizing spatio-temporal prompts to align underlying shared patterns across scenarios.

UniST achieves the above capabilities through its holistic design driven by four key components: *data*, *architecture*, *pre-training*, and *prompt learning*. Firstly, we harness the rich diversity inherent in spatio-temporal scenarios by leveraging extensive *data* from various domains and cities. Secondly, we design spatio-temporal patching to unify diverse data into a sequential format, facilitating the utilization of the powerful Transformer *architecture*. Thirdly, drawing inspiration from large language and vision models [21], [22], UniST adopts the widely-used generative *pre-training* strategy – Masked Token Modeling (MTM). We further enhance the model's capability to capture complex spatio-temporal relationships by employing multiple masking strategies that comprehensively address multi-perspective correlations. Moreover, informed by the established domain knowledge in spatio-temporal modeling, we design an innovative prompt learning approach. The elaborated prompt network identifies underlying and shared spatio-temporal patterns, adapting dynamically to generate useful prompts. In this way, UniST aligns distinct data distributions of various datasets and advances towards developing a one-for-all universal model. Our contributions are as follows:

TABLE I: Comparison of UniST with other spatio-temporal models regarding important properties.

Model	Scalability(1)	Few-shot	Zero-shot	Efficiency
PromptST [42]	✗	✗	✗	✓
GPT-ST [43]	✗	✗	✗	✓
STEP [44]	✗	✗	✗	✓
ST-SSL [36]	✗	✗	✗	✓
TrafficBERT [45]	✓	✗	✗	✓
TFM [46]	✗	✗	✗	✓
UrbanGPT [47]	✓	✓	✓	✗
STG-LLM [48]	✗	✗	✗	✗
UniST	✓	✓	✓	✓

Note: (1) The model's ability to leverage extensive datasets with diverse formats for scaling up the performance.

- To our best knowledge, we are the first to address universal spatio-temporal prediction by investigating the potential of a one-for-all model in diverse spatio-temporal scenarios.
- We propose UniST that harnesses data diversity and achieves universal spatio-temporal prediction through advanced pre-training and prompt learning. It has made a paradigm shift from traditional separate deep learning methods to a one-for-all model.
- Extensive experiments demonstrate the generality and universality of UniST. It achieves new state-of-the-art performance on various prediction tasks, particularly, superior few-shot and zero-shot capabilities.

II. Related Work

Urban Spatio-Temporal Prediction: Urban spatio-temporal prediction [4], [5] aims to model and forecast the dynamic patterns of urban activities over space and time. Deep learning techniques has propelled significant advancements. A spectrum of models, including CNNs [11], [12], RNNs [23], [24], ResNets [11], [25], MLPs [26], [27], GNNs [10], [28], [29], Transformers [30]–[32], and diffusion models [16], [33], have been introduced to capture spatio-temporal patterns. Simultaneously, cutting-edge techniques like meta-learning [34], [35], contrastive learning [36], [37], and adversarial learning [38]–[40] are also utilized. However, most approaches remain constrained by training separate models for each specific dataset. Some studies [34], [35], [41] explore transfer learning between cities, however, a certain amount of data samples in the target city are still required. Current solutions are restrictive to specified spatio-temporal scenarios and require training data, while our model allows generalization across diverse scenarios and provides a one-for-all solution.

Foundation Models for Spatio-temporal Data and Time Series: Inspired by the remarkable strides in foundation models for NLP [1], [2] and CV [18], [49], foundation models for urban prediction have emerged recently. Some explorations unlock the potential of large language models (LLMs) in this context. Intelligent urban systems like CityGPT [50], [51], CityBench [52] and UrbanGPT [47] have demonstrated proficiency in addressing language-based tasks. Additionally, LLMs are utilized for describing urban-related images [53] to benefit downstream tasks and predict user activities [54].

Moreover, the application of LLMs extends to traffic signal control [55], showcasing their utility in tackling complex spatio-temporal problems beyond languages. Recently, there also has been great progress in foundation models for time series [56]–[59]. Unlike time series characterized by a straightforward sequential structure, spatio-temporal data presents a more intricate nature with intertwined dependencies across both spatial and temporal dimensions. While exploring the integration of LLMs is promising, it's important to recognize that spatio-temporal data is not inherently generated by language. Thus, developing foundation models specifically trained on pure spatio-temporal data is also an important direction. In Table I, we compare the essential properties of UniST with other approaches employing pre-training, prompt learning, or LLMs. UniST encompasses all these essential capabilities, whereas other approaches have certain limitations.

***Prompt Learning**:* Prompt learning has achieved superior performance in large models [60], [61], with the goal of enhancing the generalization capability of pretrained models on specific tasks or domains. Typically, language models usually use a limited number of demonstrations as prompts and vision models often employ a learnable prompt network to generate useful prompts, known as prompt learning. Our research aligns with prompt learning, where spatio-temporal prompts are adaptively generated based on spatio-temporal patterns through a prompt network.

***Addressing Data Scarcity in Urban Spatio-Temporal Learning**:* Data scarcity [7], [62] poses a significant challenge in urban spatiotemporal modeling, especially when dealing with cities and regions that lack extensive historical data. Researchers have explored multiple approaches to mitigate this issue. One of the most common strategies is transfer learning [7], [41], [62]–[65], where models are pretrained on source cities with abundant data and then adapted for use in target cities with limited data. Another approach to address data scarcity is synthetic data generation [33], [66], [67], which creates additional training data by mimicking the statistical characteristics of real-world data. A more recent approach involves developing models, such as UniST [20], that can generalize across various cities without requiring city-specific training [68]. Similar to foundational models in natural language processing (NLP) and computer vision (CV) [18], [69]. UniST's "one-for-all" model approach allows it to adapt flexibly to new urban contexts while maintaining performance across a wide range of spatiotemporal conditions.

III. Methodology

A. Preliminary

Spatial and Temporal Partitions. We use a grid system for spatial partitioning, dividing the city into equal, non-overlapping areas defined by longitude and latitude on an $H \times W$ map. For each area, the temporal dynamics are recorded at certain intervals.

Spatio-Temporal Data. A spatio-temporal data X is defined as a four-dimensional tensor with dimensions $T \times C \times H \times W$. T represents time steps. C represents the number of variables, For instance, in the Taxi flow dataset, $C = 2$, where the variables are inflow and outflow. In contrast, in the Cellular dataset, $C = 1$, representing cellular network traffic. H and W represent spatial grids. T, C, H, and W can vary across different spatio-temporal scenarios.

Spatio-Temporal Prediction. For a specific dataset, given l_h historical observations for the grid map, we aim to predict the future k steps. The spatio-temporal prediction task can be formulated as learning a θ-parameterized model $\mathcal{F}$: $X_{[t:t+k]} = \mathcal{F}_\theta(X_{[t-l_h:t]})$.

Few-Shot and Zero-Shot Predictions. The model is trained on multiple source datasets and then adapted to a target dataset. In few-shot learning, it is fine-tuned with a small amount of target samples; in zero-shot learning, it makes predictions without any fine-tuning.

B. Pre-training and Prompt Learning

Universal spatio-temporal prediction aims to empower a single model to effectively handle diverse spatio-temporal scenarios, requiring the unification of varied spatio-temporal data within a cohesive model. This necessitates addressing significant distribution shifts across datasets of different scenarios. To achieve this goal, we propose a framework for pre-training and prompt learning, leading to a universal prediction model, UniST. Figure 2 shows the overview architecture, detailing UniST with two stages:

- **Stage 1: Large-scale spatio-temporal pre-training.** Different from existing methods limited to a single dataset, our approach utilizing extensive spatio-temporal data from a variety of domains and cities for pre-training.
- **Stage 2: Spatio-temporal knowledge-guided prompt learning.** We introduces a prompt network for in-context learning, where the generation of prompts is adaptively guided by well-developed spatio-temporal domain knowledge, such as spatial hierarchy and temporal periodicity.

C. Base Model

Our base model is a Transformer-based encoder-decoder architecture. Through spatio-temporal patching, it can handle diverse spatio-temporal data in a unified sequential format.

Spatio-Temporal Patching. The conventional Transformer architecture is designed for processing 1D sequential data. However, spatio-temporal data possesses a 4D structure. To accommodate this, we first split the data into channel-independent instances, which are 3D tensors. Then, we utilize spatio-temporal patching to transform the 3D tensor, denoted as $X \in \mathbb{R}^{L \times H \times W}$, into multiple smaller 3D tensors. If the original shape is $L \times H \times W$, and the patch size is (l, h, w), the resulting sequence is given by $E_x \in \mathbb{R}^{L' \times H' \times W'}, L' = \frac{L}{l}, H' = \frac{H}{h}, W' = \frac{W}{w}$.

This transformation involves a 3D convolutional layer with a kernel size and stride both set to (l, h, w). The process can be expressed as $E_x = \textsc{Conv}_{3d}(X)$, where E_x represents the converted 1D sequential data. The sequence length of E_x is $L' \times H' \times W'$.

Positional Encoding. As the original Transformer architecture does not consider the order of the sequence, we follow the common practice that incorporate positional encoding [21]. To

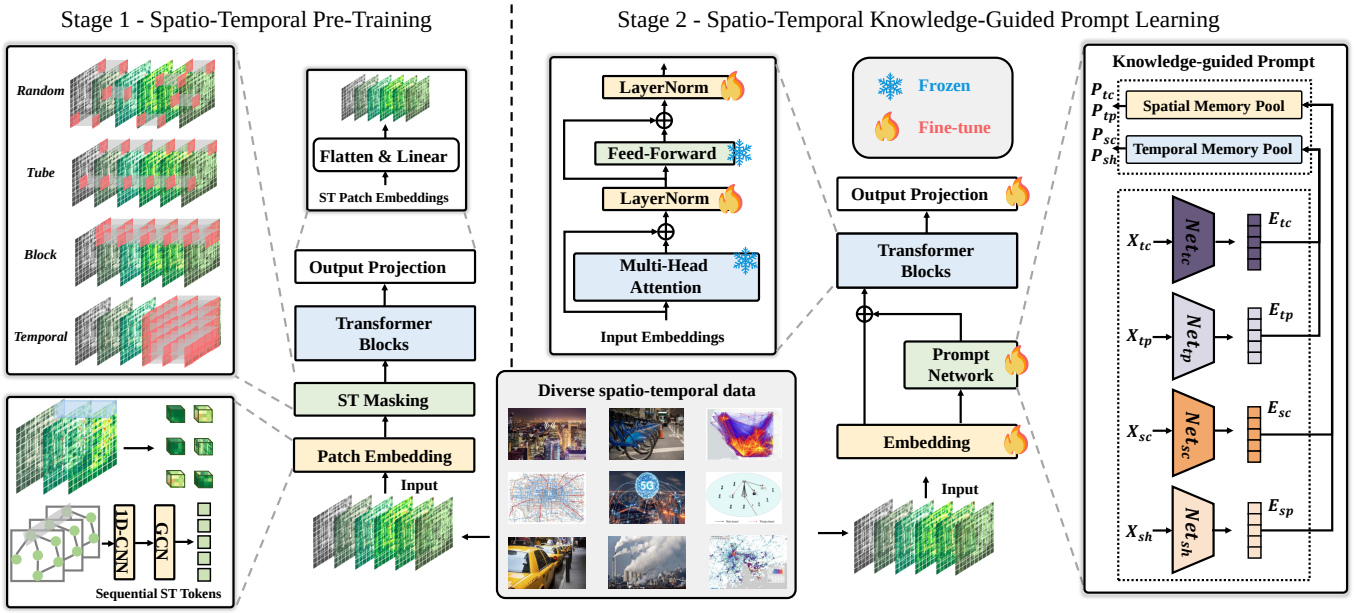


Fig. 2: The overview architecture of UniST, which consists of two stages: (i) large-scale spatio-temporal pre-training, (ii) spatio-temporal knowledge-guided prompt learning.

enhance generalization, we choose sine and cosine functions rather than learnable parameters for positional encoding. This encoding is separately applied to the two dimensions.

Encoder-Decoder Structure. The base model utilizes an encoder-decoder framework inspired by Masked Autoencoder (MAE) [22]. It processes input patches with a certain masking ratio, where the encoder takes the unmasked patches and the decoder reconstructs the image using the encoder's output and the masked patches. Our focus is on capturing comprehensive spatio-temporal dependencies, including both high-level and low-level relationships, with the goal of accurately predicting values at specific time and space coordinates. Unlike MAE, which uses a lightweight decoder for pre-training, our model employs a full-sized decoder that plays a crucial role in both pre-training and fine-tuning. It can be formulated as:

$$E_{enc} = \textsc{Encoder}(E_x),$$
$$Y_{dec} = \textsc{Decoder}(E_{enc}, E_{mask}),$$

where E_{mask} denotes embeddings of masked patches.

D. Spatio-Temporal Self-Supervised Pre-train

In pretrained language models, the self-supervised learning task is either masking-reconstruction [21] or autoregressive prediction [2]. Similarly, in vision models, visual patches are randomly masked and the pre-training objective is to reconstruct the masked pixels. To further augment the model's capacity to capture intricate spatio-temporal relationships and intertwined dynamics, we introduce four distinct masking strategies during the pre-training phase, which are shown in the left box in the stage 1 of Figure 2. Suppose the masking percentage is r, we explain these strategies as follows:

- **Random masking.** Patches are randomly masked to capture fine-grained spatio-temporal relationships.

$$M \sim \mathbf{U}[0,1],\ E_x = E_x[M < 1-r],\ M \in \mathbb{R}^{L' \times H' \times W'}.$$

- **Tube masking.** This strategy simulates scenarios where data for certain spatial units is entirely missing across all instances in time, mirroring real-world situations where some sensors may be nonfunctional—a common occurrence. The goal is to improve spatial extrapolation competence.

$$M \sim \mathbf{U}[0,1],\ E_x = E_x[:, M < 1-r],\ M \in \mathbb{R}^{H' \times W'}.$$

- **Block masking.** It involves the complete absence of an entire block of spatial units across all instances in time. The goal is to enhance spatial generalization ability.

$$M \sim \textsc{Uniform}(1,2),$$
$$E_x = E_x[:, \frac{M-1}{2}H' : \frac{M}{2}H', \frac{M-1}{2}W' : \frac{M}{2}W'].$$

- **Temporal Masking.** By masking the future data, it compels the model to reconstruct the future based solely on historical information. It refines the model's capability to capture temporal dependencies from the past to the future.

$$M = \textsc{Concat}([\mathbf{1}_{(1-r)L' \times H' \times W'}, \mathbf{0}_{rL' \times H' \times W'}]),$$
$$E_x = E_x[M = 1].$$

By employing these diverse masking strategies, the model can systematically enhance its modeling capabilities from a comprehensive perspective, simultaneously addressing spatio-temporal, spatial, and temporal relationships.

E. Spatio-Temporal Knowledge-Guided Prompt

Prompt learning plays a critical role in enhancing UniST's generalization ability. Before delving into the details of our prompt design, it is essential to discuss why pre-trained models can be applied to unseen scenarios.

*1) **Spatial-Temporal Generalization**:* In urban prediction tasks, the distributions of features and labels differ across domains and cities, denoted as $X_A \neq X_B, Y_A \neq Y_B$, where X and Y denote features and labels, while A and B represent different cities or domains. Taken A and B as a simple example, generalization involves leveraging knowledge acquired from the A dataset and adapt it to the B dataset. The key point lies in identifying and aligning "related" patterns between A and B datasets. While finding similar patterns for an entire dataset may be challenging, we claim that identifying and aligning fine-grained patterns is feasible. Specifically, we provide some assumptions that applies to prompt-empowered spatio-temporal generalization:

Assumption 1. *For a new dataset* **B***, it is possible to identify fine-grained patterns related to the training data* **A**.

$$X_A \neq X_B,\ Y_A \neq Y_B,$$
$$\exists x_a \in X_A, y_a \in Y_A,\ \exists x_b \in X_B, y_b \in Y_B, : x_a \approx x_b, y_a \approx y_b.$$

Assumption 2. *Distinct spatio-temporal patterns correspond to customized prompts.*

$$P_i^* \neq P_j^* \quad if \quad D(x_i, x_j) > \epsilon,$$
$$D(P_i^*, P_j^*) > D(P_m^*, P_n^*) \quad if \quad D(x_i, x_j) > D(x_m, x_n),$$

where x_i denotes the fine-grained spatio-temporal pattern, P_i^* represents the prompt of x_i, and D is the similarity between x_i and x_j.

Assumption 3. *There exists f_θ that captures the mapping relationship from the spatio-temporal pattern x_i to prompt P_i^*.*

$$P_i = f_\theta(x_i) \quad where \quad \theta = \underset{\theta}{argmin} \sum_i \textsc{Distance}(P_i^*, f_\theta(x_i)).$$

Based on these assumptions, our core idea is that for different inputs with distinct spatio-temporal patterns, customized prompts should be generated adaptively.

*2) **Spatio-Temporal Domain Knowledge**:* Given the aforementioned assumptions, a critical consideration is how to define the concept of "similarity" to identify and align shared spatio-temporal patterns. Here we leverage insights from well-established domain knowledge in spatio-temporal modeling [5], [11], encompassing properties related to both space and time. There are four aspects to consider when examining these properties:

- Spatial closeness: Nearby units may influence each other.
- Spatial hierarchy: The spatial hierarchical organization impacts the spatio-temporal dynamics, requiring a multi-level perception on the city structure.
- Temporal closeness: Recent dynamics affect future results, indicating a closeness dependence.
- Temporal period: Daily or weekly patterns exhibit similarities, displaying a certain periodicity.

For simplicity, we provide some straightforward implementations, which are shown in the four networks in Figure 2, *i.e.,* $\textsc{Net}_{tc}$, $\textsc{Net}_{tp}$, $\textsc{Net}_{sc}$, and $\textsc{Net}_{sh}$. For the spatial dimension, we first employ an attention mechanism to merge the temporal dimension into a representation termed E_s. Then,

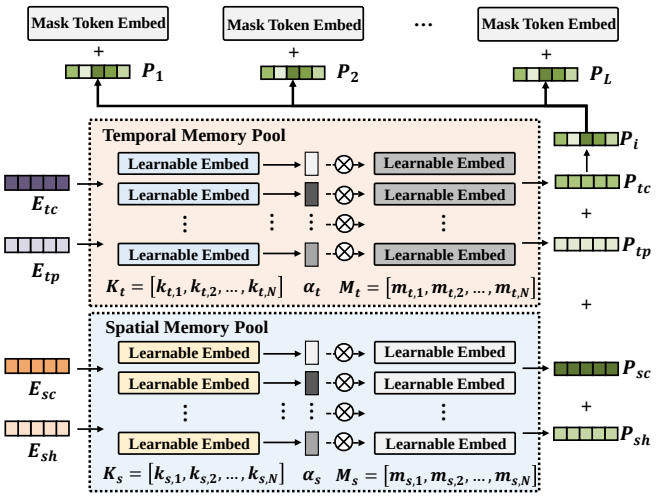


Fig. 3: Illustration of the prompt generation process.

to capture spatial dependencies within close proximity, a two-dimensional convolutional neural network (CNN), *i.e.,* $\textsc{Net}_{sc}$, with a kernel size of 3 is employed. To capture spatial hierarchies, we utilize CNNs with larger kernel sizes, *i.e.,* $\textsc{Net}_{sh}$. These larger kernels enable the perception of spatial information on larger scales, which facilitate to construct a hierarchical perspective. As for the temporal dimension, we employ an attention network, *i.e.,* $\textsc{Net}_{tc}$, to aggregate the previous M steps denoted as X_c. Regarding the temporal period, we select corresponding time points from the previous N days, denoted as X_p. Subsequently, we employ another attention network, *i.e.,* $\textsc{Net}_{tp}$, to aggregate the periodical sequence, which captures long-term temporal patterns. The overall process is formulated as follows:

$$E_{sc} = \textsc{Conv}_{2D}[3](X_s),$$
$$E_{sh} = \{\textsc{Conv}_{2D}[2^i + 1](X_s)\}, i \in \{2, 3, 4\},$$
$$E_{tc} = \textsc{Attention}(X_c),$$
$$E_{tp} = \textsc{Attention}(X_p).$$

It is essential to emphasize that the learning of E_{sc}, E_{sh}, E_{tc}, and E_{tp} is not restricted by our practice. Practitioners have the flexibility to employ more complex designs to capture richer spatio-temporal properties. For example, Fourier-based approaches [70], [71] can be utilized to capture periodic patterns.

*3) **Spatio-Temporal Prompt Learner**:* Given the representations of properties derived from spatio-temporal domain knowledge, the pivotal question is how to generate prompts—*how does spatio-temporal knowledge guide prompt generation?* Here we utilize prompt learning techniques. While prompt learning in computer vision [61] often train fixed prompts for specific tasks such as segmentation, detection, and classification. Due to the high-dimensional and complex nature of spatio-temporal patterns, training a fixed prompt for each case becomes impractical.

To tackle this issue, we draw inspirations from memory networks [72] and propose a novel approach that learns a spatial memory pool and a temporal memory pool. In the

prompt learning process, these memory pools are optimized to store valuable information about spatio-temporal domain knowledge. As shown in Figure 3, the spatial and memory pools are defined as follows:

$$KM_s = \{(k_{s,0}, m_{s,0}), (k_{s,1}, m_{s,1}), ..., (k_{s,N-1}, m_{s,N-1})\},$$
$$KM_t = \{(k_{t,0}, m_{t,0}), (k_{t,1}, m_{t,1}), ..., (k_{t,N-1}, m_{t,N-1})\},$$

where $k_{s,i}, m_{s,i}, k_{t,i}, m_{t,i}, i \in \{0, 1, ..., N-1\}$ are all learnable parameters, and the memory is organized in a key-value structure following existing practice [72], [73].

Subsequently, useful prompts are generated based on these optimized memories. This involves using the representations of spatio-temporal properties as queries to extract valuable memory knowledge, *i.e.,* pertinent embeddings from the memory pool. Figure 3 illustrates the process, and it is formulated as follows:

$$\alpha_{sc} = [k_{s,0}; k_{s,1}; ..., k_{s,N-1}]E_{sc}^T, \ P_{sc} = \sum_i \alpha_{sc,i} m_{s,i},$$
$$\alpha_{sh} = [k_{s,0}; k_{s,1}; ..., k_{s,N-1}]E_{sh}^T, \ P_{sh} = \sum_i \alpha_{sh,i} m_{s,i},$$
$$\alpha_{tc} = [k_{t,0}; k_{t,1}; ..., k_{t,N-1}]E_{tc}^T, \ P_{tc} = \sum_i \alpha_{tc,i} m_{t,i},$$
$$\alpha_{tp} = [k_{t,0}; k_{t,1}; ..., k_{t,N-1}]E_{tp}^T, \ P_{tp} = \sum_i \alpha_{tp,i} m_{t,i},$$

where $E_{sc}, E_{sh}, E_{tc}, E_{tp}$ represent four representations related to four types of spatio-temporal domain knowledge, and $P_{sc}, P_{sh}, P_{tc}, P_{tp}$ are the extracted prompts. This allows the model to adaptively select the most useful information for prediction. These prompts are then integrated into the input space of the Transformer architecture, which are displayed in the upper part of Figure 3.

F. UniST for Graph-based Data

UniST is quite versatile and can handle various types of structural data. Besides grid-based data, it can also process graph-based spatio-temporal data, where spatial units are organized according to a graph structure. This organization is common in spatio-temporal prediction, as city road maps often serve as the topology. Graph-based spatio-temporal data are often described as $\mathcal{G}_{ST} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{X})$, where (1) $\mathcal{V} = v_1, v_2, \ldots, v_N$ denotes the set of nodes, with N being the number of nodes. (2) $\mathcal{E} = e_{ij} = (v_i, v_j)$ denotes the set of edges. (3) $\mathcal{A}$ is the adjacency matrix. (4) $\mathcal{X}$ represents the time series data for each node. The spatio-temporal data for a graph structure can be represented as $X \in \mathbb{R}^{T \times N \times 1}$.

To adapt UniST to graph-based data, we adjusted the spatial patch size to 1 and applied the same spatio-temporal patching approach used for grid-based data. This allows us to adapt a pretrained UniST model from grid-based data to graph-based data by simply training an additional patching encoder.

G. Model Training

In training UniST, we utilize mean squared error (MSE) as the loss function, which calculates the error between the reconstructed masked portions and the ground truth values. The MSE loss is formulated as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (Y_{dec}^i - \hat{Y^i})^2$$

where Y_{dec} represents the output of the decoder, Y denotes the ground truth, and N is the number of masked samples. This loss function guides the model in minimizing the error between predictions and actual values, thereby improving its reconstruction accuracy.

The training process alternates across multiple datasets and employs four distinct masking strategies. During each iteration, we randomly select a dataset and a masking strategy, enabling the model to perform gradient descent training under diverse conditions. This method strengthens the model's robustness by exposing it to various spatiotemporal scenarios, while also reducing the risk of overfitting by ensuring the model learns from a broad set of inputs and objectives. Let $D = \{D_1, D_2, \ldots, D_m\}$ represent the datasets available for training, and $M = \{M_1, M_2, \ldots, M_4\}$ denote the set of masking strategies. We define $\mathcal{L}(d_i, m_i)$ as the loss function based on the selected dataset d_i and masking strategy m_i, with θ representing the model parameters. The complete training process can be outlined as follows:

$$\begin{aligned} &\text{For } i = 1 \text{ to } N: \\ &\quad d_i \sim \text{Uniform}(D), \quad m_i \sim \text{Uniform}(M) \\ &\quad \Rightarrow \quad \theta \leftarrow \theta - \eta \nabla \mathcal{L}(d_i, m_i; \theta) \end{aligned}$$

where N represents the total training iterations, and η is the learning rate.

H. Method Discussion

UniST effectively addresses complex spatio-temporal dependencies through several key mechanisms. First, By employing a self-attention mechanism across the unified spatio-temporal dimensions, the model gains the flexibility to learn complex dependencies. This design allows it to identify relationships between any points in time and space without being limited to predefined dimensions. Second, the integration of spatio-temporal knowledge-guided prompts within the Transformer framework enhances the model's ability to tailor its predictions. These prompts facilitate adaptive responses based on the unique characteristics of each spatio-temporal scenario, allowing the model to manage dependencies more effectively. Last but not least, the scalability of the Transformer architecture enables it to handle spatio-temporal data of various forms. This characteristic supports pre-training on a diverse array of datasets, which helps the model to learn the underlying structures and dependencies inherent in urban environments comprehensively.

TABLE II: Performance comparison of short-term prediction on seven datasets in terms of MAE and RMSE. We use the average prediction errors over all prediction steps. Bold denotes the best results and underline denotes the second-best results.

	TaxiBJ		Crowd		Cellular		BikeNYC		TrafficJN		TDrive		TrafficSH	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
HA	53.77	29.82	17.80	6.79	72.94	27.57	11.41	3.43	1.38	0.690	150.2	74.5	1.24	0.771
ARIMA	56.70	39.53	21.87	10.23	81.31	40.22	12.37	3.86	1.20	0.651	211.3	108.5	1.17	0.769
STResNet	45.17	30.87	5.355	3.382	24.30	14.32	8.20	4.98	0.964	0.556	220.1	117.4	1.00	0.723
ACFM	37.77	21.59	4.17	2.34	22.79	12.00	3.93	1.67	0.920	0.559	98.1	51.9	0.833	0.566
STID	27.36	14.01	3.85	1.63	18.77	8.24	4.06	1.54	0.880	0.495	47.4	23.3	0.742	0.469
STNorm	29.37	15.71	4.44	2.09	19.77	8.19	4.45	1.66	0.961	0.532	54.3	47.9	0.871	0.579
STGSP	45.04	28.28	7.93	4.56	39.99	21.40	5.00	1.69	0.882	0.490	94.6	47.8	1.02	0.749
MC-STL	29.14	15.83	4.75	2.39	21.22	10.26	4.08	2.05	1.19	0.833	54.2	28.1	1.00	0.720
PromptST	27.44	14.54	3.52	1.54	15.74	7.20	4.36	1.57	0.953	0.490	47.5	22.8	0.811	0.523
MAU	38.14	20.13	4.94	2.35	39.09	18.73	5.22	2.06	1.28	0.697	48.8	22.1	1.37	0.991
PredRNN	27.50	14.29	5.13	2.36	24.15	10.44	5.00	1.74	0.852	0.463	54.9	25.2	0.748	0.469
MIM	28.62	14.77	5.66	2.27	21.38	9.37	4.40	1.62	1.17	0.650	51.4	22.7	0.760	0.505
SimVP	32.66	17.67	3.91	1.96	16.48	8.23	4.11	1.67	0.969	0.556	46.8	22.9	0.814	0.569
TAU	33.90	19.37	4.09	2.11	17.94	8.91	4.30	1.83	0.993	0.566	51.6	28.1	0.820	0.557
PatchTST	42.74	22.23	10.25	3.62	43.40	15.74	5.27	1.65	1.25	0.616	106.4	51.3	1.10	0.663
iTransformer	36.97	19.14	9.40	3.40	37.01	13.93	7.74	2.53	1.11	0.570	86.3	42.6	1.04	0.655
PatchTST(one-for-all)	43.66	23.16	13.51	5.00	56.80	20.56	9.97	3.05	1.30	0.645	127.0	59.26	1.13	0.679
UniST(one-for-all)	**26.84**	**13.95**	**3.00**	**1.38**	**14.29**	**6.50**	**3.50**	**1.27**	**0.843**	**0.430**	**44.97**	**19.67**	**0.665**	**0.405**

IV. Performance Evaluations

A. Experimental Setup

To evaluate the performance of UniST, we conducted extensive experiments on more than 20 spatio-temporal datasets.

Datasets. The datasets we used cover multiple cities, spanning various domains such as crowd flow, dynamic population, traffic speed, cellular network usage, taxi trips, and bike demand. Appendix Table VIII and Table IX provide a summary of the datasets we used. These spatio-temporal datasets originate from distinct domains and cities, and have variations in the number of variables, sampling frequency, spatial scale, temporal duration, and data size.

Baselines. We compare UniST with a broad collection of state-of-the-art models for spatio-temporal prediction, which can be categorized into five groups:

- **Heuristic approaches.** History average (HA) and ARIMA.
- **Deep urban prediction approaches.** We consider state-of-the-art urban ST prediction models, including STResNet [11], ACFM [12], MC-STL [37], STGSP [74], STNorm [75], STID [27], and PromptST [42].
- **Foundational models for traffic prediction.** Given that UniST is designed as a universal model, we selected state-of-the-art foundational models for traffic prediction that leverage self-supervised learning to ensure a fair comparison. These models include GPT-ST [43], STEP [44], ST-SSL [36], and an urban foundation model, UrbanGPT [47]. Although GPT-ST, STEP, and ST-SSL are termed foundational models, they are limited to training on data with fixed node structures, restricting them to datasets with a fixed number of nodes. For UrbanGPT, we used the pretrained model on multiple datasets and evaluated its performance in zero-shot scenarios.
- **Video prediction approaches.** We compare with competitive video prediction models from the popular benchmark, including PredRNN [23], MAU [76], MIM [77], SimVP [78], and TAU [79].
- **Multivariate time series forecasting approaches.** We consider state-of-the-art multivariate time series forecasting models, including PatchTST [80] and iTransformer [81]. For a fair comparison, we also train PatchTST for all datasets, denoted as PatchTST(one-for-all).
- **Meta learning approaches.** To evaluate the generalization capability, we consider meta-learning approaches including MAML [82] and MetaST [35].

Metrics. We employed commonly used regression metrics, including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). For more detailed information of the datasets, baselines, and metrics, please refer to Appendix A, Appendix B, and Appendix D.

Parameter Settings. Table X shows the parameter details of UniST with different sizes. During the training process, we used the Adam optimizer for gradient-based model optimization. The learning rate of the pre-training is set as 3e-4, and the learning rate of the prompt tuning is set as 5e-5. The pre-training learning rate is selected via grid searching in a set of $\{1e-3, 3e-4, 1e-4\}$, and the fine-tuning learning rate is selected in a set of $\{1e{-}4, 5e{-}5, 1e{-}5\}$. Both in pre-training and fine-tuning, we evaluate the model's performance on the validation set every ten epochs ($\sim$all training instances). We choose the model that performs best on the validation set for evaluations on the testing set.

B. Short-Term Prediction

Setups: Following previous practices [58], [80], both the input step and prediction horizon are set as 6, i.e., $6 \rightarrow 6$. For baselines, we train a dedicated model for each dataset, while UniST is evaluated across all datasets.

Results: Table II presents the short-term prediction results, with a selection of datasets due to space constraints. The complete results can be found in Table XII and Table XIII in Appendix E. As we can observe from Table II, UniST consistently outperforms all baselines across all datasets. Compared with the best baseline of each dataset, it showcases a notable average improvement. Notably, time series approaches such as PatchTST and iTransformer exhibit inferior performance compared to spatio-temporal methods. This underscores the importance of incorporating spatial dependency as prior knowledge for spatio-temporal prediction tasks. Another observation is that PatchTST(one-for-all) performs worse than PatchTST

TABLE III: Performance comparison of long-term prediction on three datasets. We use the average prediction errors over all prediction steps. Bold denotes the best results and underline denotes the second-best results.

	TaxiNYC		Crowd		BikeNYC	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
HA	61.03	21.33	19.57	8.49	11.00	3.66
ARIMA	68.0	28.66	21.34	8.93	11.59	3.98
STResNet	29.54	14.46	8.75	5.58	7.15	3.87
ACFM	32.91	13.72	6.16	3.35	4.56	1.86
STID	24.74	11.01	4.91	2.63	4.78	2.24
STNorm	31.81	11.99	9.62	4.30	6.45	2.18
STGSP	28.65	10.38	17.03	8.21	4.71	1.54
MC-STL	29.29	17.36	9.01	6.32	4.97	2.61
MAU	26.28	9.07	20.13	8.49	6.18	2.13
PredRNN	21.17	7.31	19.70	10.66	5.86	1.97
MIM	63.36	29.83	15.70	8.81	7.58	2.81
SimVP	20.18	9.78	5.50	3.13	4.10	1.71
TAU	24.97	10.93	5.31	2.81	3.89	1.73
PatchTST	30.64	17.49	5.25	2.83	5.27	1.65
iTransformer	33.81	11.48	6.94	2.63	6.00	2.02
PatchTST(one-for-all)	34.50	10.63	6.39	2.92	6.02	1.83
UniST (one-for-all)	**19.83**	**6.71**	**4.25**	**2.26**	**3.56**	**1.31**

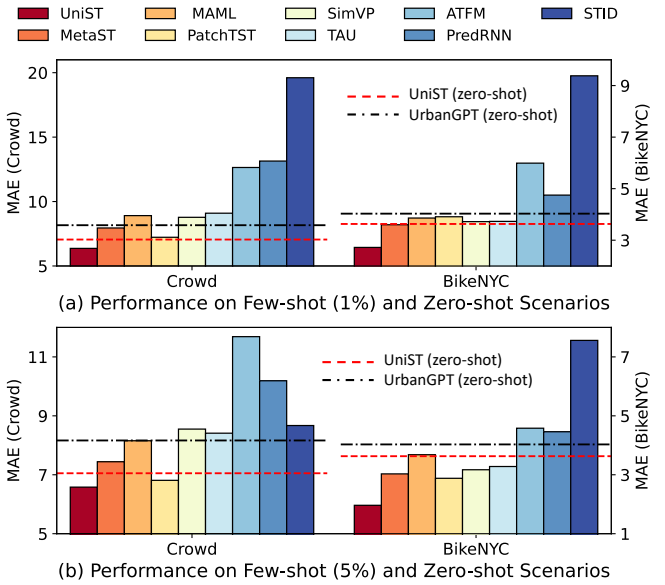


Fig. 4: (a) Few-shot performance of UniST and baselines on Crowd and BikeNYC datasets using only 1% of the training data. (b) Few-shot performance of UniST and baselines using only 5% of the training data. The Dashed red lines denote the zero-shot performance of UniST.

dedicated for each dataset, suggesting that the model struggles to directly adept to these distinct data distributions. Moreover, baseline approaches exhibit inconsistent performance across diverse datasets, indicating their instability across scenarios. The consistent superior performance of UniST across all scenarios underscores the significant potential and benefits of a one-for-all model. Moreover, it demonstrates UniST's capability to orchestrate diverse data, where different datasets can benefit each other.

C. Long-Term Prediction

Setups: Here we extend the input step and prediction horizon to 64 following [58], [80]. This configuration accommodates prolonged temporal dependencies, allowing us to gauge the model's proficiency in capturing extended patterns over time. Similar to the short-term prediction, UniST is directly evaluated across all datasets, while specific models are individually trained for each baseline on respective datasets.

Results: Table III shows the long-term prediction results. Even with a more extended prediction horizon, UniST still consistently outperforms all baseline approaches across all datasets. Compared with the best baseline of each dataset, it yields an average improvement of 10.1%. This highlights UniST's capability to comprehend temporal patterns effectively and its robustness in generalizing across extended durations. Table XIV in Appendix E illustrates the complete results.

D. Few-Shot Prediction

Setups: The hallmark of large foundation models lies in their exceptional generalization ability. The few-shot and zero-shot evaluations are commonly employed to characterize the ultimate tasks for universal time series forecasting [7], [59]. Likewise, the few-shot and zero-shot prediction capability is crucial for a universal spatio-temporal model. In this section, we assess the few-shot learning performance of UniST. Each dataset is partitioned into three segments: training data, validation data, and test data. In few-shot learning scenarios, when confronted with an unseen dataset during the training process, we utilized a restricted amount of training data, specifically, *1%, 5%, 10%* of the training data. We choose some baselines with relatively good performance for the few-shot setting evaluation, We also compare with meta-learning baselines, *i.e.,* MAML and MetaST, and pretraining and finetuning-based time series method, *i.e.,* PatchTST.

Results: Appendix Table XV to Table XVII illustrate the overall few-shot results. Due to the space limit, Figure 4 only illustrates the 1% few-shot learning results on two datasets. In these cases, UniST still outperforms all baselines, it achieves a larger relative improvement over baselines compared to long-term and short-term predictions. The transferability can be attributed to successful knowledge transfer in our spatio-temporal prompt.

E. Zero-Shot Prediction

Setups: Zero-shot inference is a critical task for evaluating foundation models' generalization capabilities, as it assesses performance on entirely new datasets without prior exposure. In this context, after UniST is trained on a diverse collection of datasets, we evaluate its performance on a completely novel dataset with no prior training data. The test data used in this scenario aligns with that of normal prediction and few-shot prediction. Specifically, we include UrbanGPT [47] as a competitive zero-shot baseline for comparison.

Results: Figure 4 presents a comparison between UniST (in zero-shot mode) and baseline models in few-shot mode, as well as a direct comparison with UrbanGPT [47]. As shown, UniST achieves impressive zero-shot performance, even surpassing many baselines that were trained with access

TABLE IV: Performance comparision on two graph-based datasets, e.g., METR-LA and Crowd-BJ datasets.

Model	METR-LA		Crowd-BJ	
	MAE	RMSE	MAE	RMSE
STGCN	3.01	6.61	3.04	4.89
TrafficBert	3.53	5.72	3.16	4.11
STG-LLM	2.81	5.90	3.01	4.42
STNorm	2.98	6.23	3.12	4.76
GPT-ST	3.08	6.37	2.71	4.12
STEP	3.37	6.99	2.94	4.36
ST-SSL	2.80	5.58	2.78	4.15
STID	3.40	6.72	2.79	4.23
UniST(train from scratch)	2.56	6.19	2.45	3.68
UniST(finetune from grid data)	2.05	5.26	2.42	3.55

to the target dataset, indicated by red dashed lines. Notably, UniST also outperforms UrbanGPT in zero-shot scenarios, highlighting its robust generalization capabilities. We attribute these surprising results to the powerful spatio-temporal transfer capability. It suggests that for a completely new scenario, even when the displayed overall patterns are dissimilar to the data encountered during the training process, UniST can extract fine-grained similar patterns from our defined spatial and temporal properties. These findings underscore UniST's superior generalization ability in both few-shot and zero-shot scenarios, demonstrating its potential for real-world applications in diverse and unseen contexts.

F. Prediction on Graph Data

***Setups**:* The above prediction experiments use only grid-based data, and it is notable that UniST has the ability to generalize to graph-based data, which is very common in the field of spatio-temporal modeling. It is important to unify these two types of data formats. We evaluated UniST on two widely used spatio-temporal graph datasets, METR-LA [7], [83] and Crowd-BJ [7], [33], alongside state-of-the-art baselines including STGCN [84], STNorm [75], and STID [27]. We also compare with foundational models for traffic prediction, including GPT-ST [43], STEP [44], and ST-SSL [36]. For the graph-based data experiments, we tested UniST in two ways: (1) training from scratch using graph-based data, (2) fine-tuning a model pre-trained with grid-based data.

***Results**:* Table IV shows the comparison results on graph-based data. We observe that UniST outperformed the baselines significantly. UniST finetuned from grid data achieves an average improvement of 19.2%. These results suggest its capability to generalize to graph-based data. The performance of UniST trained from scratch indicates that even within a single dataset, our elaborated knowledge-guided prompts facilitate learning useful patterns, which are then leveraged for prediction. Additionally, the benefits of fine-tuning from grid data highlight UniST's effectiveness in capturing universal spatio-temporal patterns shared between grid-based and graph-based scenarios.

The inclusion of graph-based data further validates the universality of UniST, showcasing its adaptability to diverse ST scenarios. This expansion highlights our overall framework, including the detailed pre-training and fine-tuning processes, are highly versatile and can be flexibly applied to different data formats. The promising results suggest that with more data, UniST can further improve its capabilities and expand its applicability across a wider range of tasks.

TABLE V: Ablation studies on four masking strategies.

	Prediction	Imputation	Spatial extrapolation
Complete	0.781	0.761	0.729
wo/ Random masking	0.796	1.72	0.761
wo/ Tube masking	0.787	0.788	0.817
wo/ Block masking	0.785	0.773	1.02
wo/ Temporal masking	1.44	0.772	0.742

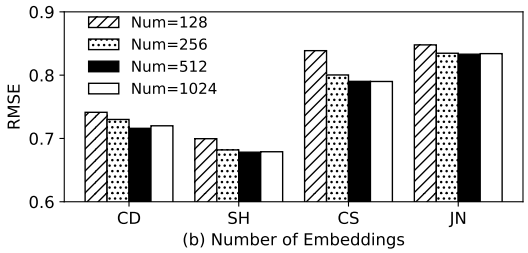


Fig. 5: Ablation studies of varying the number of learnable embeddings in the temporal and spatial memory pools on four traffic speed datasets: Chengdu (CD), Shanghai (SH), Changsha (CS), and Jinan (JN).

V. Study and Analysis on UniST

A. Ablation Study

1) Masking Strategies: We investigated the contribution of each of the four masking strategies by comparing the performance when all four strategies are employed with the performance when one of the strategies is removed. We conducted experiments on three spatio-temporal tasks: prediction, imputation, and spatial extrapolation, using the TrafficCD dataset. Prediction involves forecasting future values based on historical data, imputation focuses on filling missing values in incomplete data sequences, and spatial extrapolation aims to predict values at unobserved spatial locations. We use Root Mean Squared Error (RMSE) as the evaluation metric to assess performance across these tasks.

The results, shown in Table V, indicate that training with all four masking strategies achieved the best performance across all three tasks. Removing the temporal masking strategy results in the most significant performance decrease for the prediction task, removing the random masking strategy leads to the most significant performance decrease for the imputation task, and removing the block masking strategy results in the most significant performance decrease for the spatial extrapolation task. These results are reasonable as each masking strategy is designed to align with a specific task objective.

It is worth noting that despite the seemingly mismatched nature of some masking strategies with certain spatio-temporal tasks (e.g., random masking vs. prediction, temporal masking vs. imputation, and temporal masking vs. spatial extrapolation), we find that these masking strategies still contribute to the performance of less related tasks. This indicates that the masking strategies not only benefit their intended tasks but also

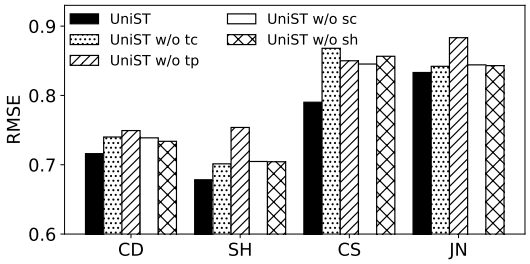


Fig. 6: Ablation studies on four types spatial and temporal knowledge extraction t_c, t_p, s_c, and s_h.

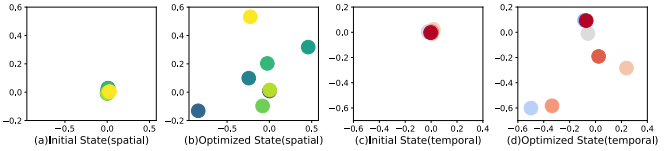


Fig. 7: Embeddings visualization of spatial and temporal memory pools at the initial and final optimized states. The embeddings exhibit obvious divergence.

have broader effects on the model's general learning of spatio-temporal dependencies and dynamics. For example, while random masking may seem unrelated to causal prediction tasks, it can help the model learn robust features that generalize well across different time points. Additionally, temporal masking can help the model better understand the temporal dynamics when performing spatial extrapolation.

2) Knowledge-Guide Prompts: The prompts play an essential role in our UniST model. Here we investigate whether the designed spatial and temporal properties s_c, s_h, t_c, and t_p contribute to the final performance. We use s_c to denote spatial closeness, s_h to denote spatial hierarchy, t_p for temporal periodicity, and t_c for temporal closeness.

we compare the overall design that incorporates all four properties with four degraded versions that individually remove s_c, s_h, t_c, or t_p . Figure 6 shows the results on four traffic speed datasets. As we can observe, removing any property results in a performance decrease. The contributions of each spatial and temporal property vary across different datasets, highlighting the necessity of each property for the spatio-temporal design.

Additionally, we explore how the number of memory units (embeddings) in the memory pool affects the model's performance, as shown in Figure 5. Here, the "number of embeddings" specifically refers to the distinct memory units that capture typical patterns for the model to extract, store, and retrieve as needed, not the traditional embedding dimensions. Increasing the memory units from 128 to 512 enhances performance across all four datasets. Further increasing the number to 1024 yields similar results to 512, suggesting that 512 is the optimal choice.

B. Prompt Learner

In this section, we conduct in-depth analyses of the prompt learner. To provide a clearer understanding, we leverage t-Distributed Stochastic Neighbor Embedding (t-SNE) to visualize the embeddings of both the spatial and temporal memory

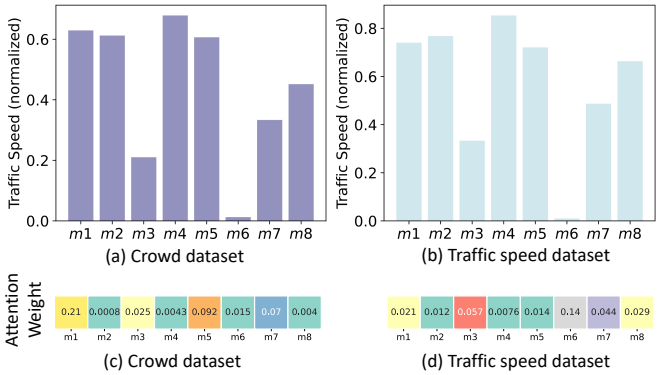


Fig. 8: (a) and (b): Comparison of the mean value of inputs in each memory embedding, where the inputs assign the highest attention weight to the memory embedding. (c) and (d): Comparison of the attention weight on each memory embedding for two distinct datasets.

pools. Specifically, we plot the initial state and the optimized state in Figure 7. Notably, from the start state to the final optimized state, the embeddings gradually become diverged in different directions. This suggests that, throughout the optimization process, the memory pools progressively store and encapsulate personalized information.

Next, we delve into the memorized patterns of each embedding within the temporal memory pool. Specifically, we first select the inputs based on the attention weights. For each embedding, we aggregate the corresponding input spatio-temporal data with the highest attention weight. Then, we calculate the mean value of the extracted spatio-temporal data. Figure 8(a) and Figure 8(b) illustrate the results for two datasets (Crowd and TrafficSH). As we can see, the memorized patterns revealed in the prompt tool exhibit remarkable consistency across different urban scenarios. This not only affirms that each embedding is meticulously optimized to memorize unique spatio-temporal patterns, but also underscores the robustness of the spatial and temporal memory pools across different scenarios.

Moreover, we examine the extracted spatio-temporal prompts for two distinct domains. Specifically, we calculate the mean attention weight for each embedding in the context of each dataset. Figure 8(c) and Figure 8(d) illustrate the comparison results. As we can observe, the depicted attention weight distributions for the two datasets manifest striking dissimilarities. The observed distinctiveness in attention weight distributions implies a dynamic and responsive nature in the model's ability to tailor its focus based on the characteristics of the input data. The ability to dynamically adjust the attention weights reinforces UniST's universality.

C. Scalability

Since UniST is a unified model trained on diverse and extensive data, understanding its scalability in relation to data size and model capacity is crucial. Conducting ablation studies with varying data sizes and model parameters not only reveals the model's adaptability to resource constraints but also offers valuable insights for practitioners seeking to

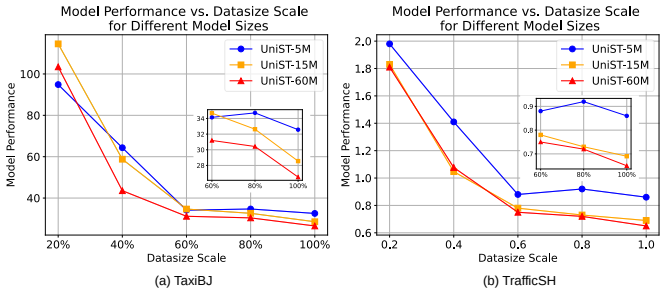


Fig. 9: Scalability of UniST with different model sizes.

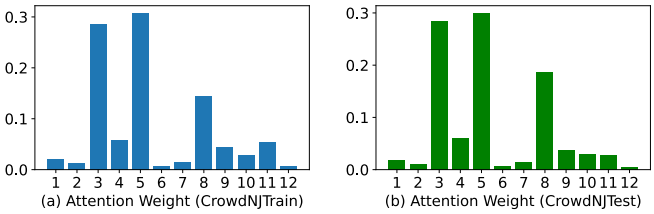


Fig. 10: Comparison of attention weight distribution between the training set and testing set of the CrowdNJ dataset.

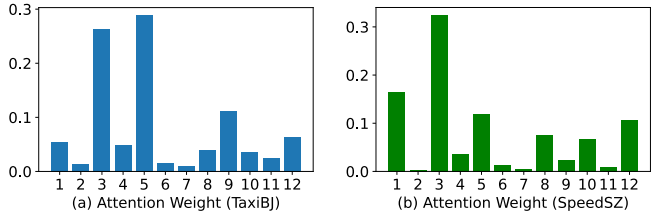


Fig. 11: Comparison of attention weight distribution between the the TaxiBJ dataset and SpeedSZ dataset.

develop foundation models. In our experiments, we investigate the relationship between model performance, data scale, and model size, as illustrated in Figure 9. We evaluated three model configurations: small (∼5M parameters), medium (∼15M parameters), and large (∼60M parameters), relative to dataset sizes (20%, 40%, 60%, 80%, and 100%).

Key observations reveal that larger models exhibit faster initial convergence during early training stages and maintain a performance edge as data scales increase. Notably, from 60% to 100% of the dataset, larger models demonstrate a steeper performance improvement, suggesting enhanced scalability with data. While we classify the 60M parameter model as "large" in our context, it's important to acknowledge that it is modest compared to the scale of large language or vision models [18], [69]. However, even at this scale, the model shows remarkable generalization capabilities and universality. While the current dataset size is insufficient to fully capture scaling laws, our approach shows significant potential for further development as larger datasets become available.

D. Analysis of Distribution Shifts

In real-world applications of prediction, spatio-temporal data often differ significantly in their patterns due to factors such as location, domain, and data collection conditions. For

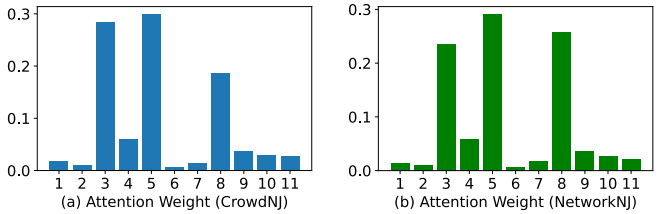


Fig. 12: Comparison of attention weight distribution between the the CrowdNJ dataset and the CellularNJ dataset.

a universal model like UniST, which is designed to generalize across diverse datasets, effectively managing distribution shifts is crucial to maintaining robust performance. Prompt-based learning provides a flexible mechanism for adapting to these shifts by dynamically adjusting model behavior based on the unique characteristics of each dataset.

To gain insights into how UniST handles distribution shifts, we conduct a detailed analysis of the generated prompts across different datasets. By examining attention weights on the embeddings within the memory pool and visualizing their distribution, we can understand how UniST captures and adapts to variations in data distribution. For each dataset, we compute the attention weights on the embeddings in the memory pool and visualize the distribution of these weights in Figures 10 to 12. Specifically, the attention weight analysis is based on the model whose performance is evaluated in Table II. We have selected three typical scenarios to explore:

1) **Training and Testing Sets of One Dataset:** This analysis aims to investigate the model's ability to generalize within a familiar dataset.
2) **Two Datasets from Different Domains in the Same City:** Understanding how the model adapts its prompt generation across different but related datasets can provide insights into its domain-specific learning.
3) **Datasets from Different Cities and Domains:** This scenario highlights the model's ability to leverage knowledge learned previously and generate useful prompts adaptively.

As shown in Figures 10 to 12, our analysis reveals compelling insights into the effectiveness of our prompting mechanism in handling distribution shifts. Specifically, we observed that similar prompts are consistently generated for datasets exhibiting similar spatio-temporal patterns. For instance, the prompts generated for the training and testing sets of a single dataset, as well as for the testing sets of two datasets from different domains within the same city, are similar. This consistency in prompt generation suggests that our model effectively captures and leverages the underlying spatio-temporal patterns shared between these datasets. Meanwhile, our model generates distinct prompts for scenarios involving datasets from different cities and domains, indicating its ability to adapt to diverse spatio-temporal contexts. This adaptability is crucial for handling distribution shifts, as it allows the model to flexibly adjust its prompt generation strategy based on the unique characteristics of each dataset.

TABLE VI: Performance on different noise levels with sine-cosine positional encoding.

Noise level	TaxiBJ	Crowd	Cellular	BikeNYC	TrafficSH
0	26.841	3.00	14.294	3.506	0.6650
0.1%	26.846	3.038	14.297	3.507	0.6651
1%	26.90	3.039	14.390	3.534	0.6653
10%	28.76	3.29	14.91	3.695	0.6877
Best baseline	27.36	3.85	16.48	3.93	0.742

TABLE VII: Performance on different noise levels with learnable positional encoding.

Noise level	TaxiBJ	Crowd	Cellular	BikeNYC	TrafficSH
0	27.02	3.31	15.054	3.609	0.686
0.1%	27.032	3.310	15.068	3.607	0.6860
1%	27.29	3.589	16.544	3.696	0.6911
10%	43.80	11.436	70.360	8.173	1.228
Best baseline	27.36	3.85	16.48	3.93	0.742

E. Performance under Noise Perturbations

The model's ability to handle noisy data is necessary to ensure reliable predictions. Therefore, we conduct experiments to evaluate UniST's robustness against noisy data. Specifically, we introduced Gaussian noise with varying levels of intensity to the input data and assessed UniST's performance under these conditions. We considered three levels of noise: Gaussian noise randomly sampled from a 0.1% normal distribution, Gaussian noise randomly sampled from a 1% normal distribution, and Gaussian noise randomly sampled from a 10% normal distribution. These noise levels represent varying degrees of data corruption, simulating real-world scenarios where data can be noisy or contain irregularities.

The results, as detailed in Table VI, demonstrate that UniST consistently outperforms baseline models even in the presence of noise perturbations (where the best baseline has no noise perturbation). This suggests that UniST is capable of effectively handling noisy data, which is crucial for ensuring reliable predictions, especially in real-world scenarios where data can be messy or contain irregularities.

Moreover, we examine how different positional encoding methods affect the model's robustness. We compare the use of two positional encoding methods: learnable embeddings and sine-cosine encoding. The results in Table VII show the performance with learnable embeddings, while Table VI shows the performance with sine-cosine encoding. Comparing these two sets of results, we observe that sine-cosine encoding exhibits more robust performance against noise perturbations. Specifically, learnable embeddings show a significant performance reduction with increased noise perturbation and perform worse than the best baseline model.

VI. Conclusion

In this work, we address an important problem of building a universal model UniST for urban spatio-temporal prediction. By leveraging the diversity of spatio-temporal data from multiple sources, and discerning and aligning underlying shared spatio-temporal patterns across multiple scenarios, UniST demonstrates a powerful capability to predict across all scenarios, particularly in few-shot and zero-shot settings. A promising direction for future work entails the integration of various spatio-temporal data formats, such as grid, sequence, and graph data. Our study inspires future research in spatio-temporal modeling towards the universal direction.

References

[1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[3] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[4] S. Wang, J. Cao, and S. Y. Philip, "Deep learning for spatio-temporal data mining: A survey," *IEEE transactions on knowledge and data engineering*, vol. 34, no. 8, pp. 3681–3700, 2020.

[5] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 3, pp. 1–55, 2014.

[6] J. Gong, Y. Liu, T. Li, H. Chai, X. Wang, J. Feng, C. Deng, D. Jin, and Y. Li, "Empowering spatial knowledge graph for mobile traffic prediction," in *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 2023, pp. 1–11.

[7] Y. Yuan, C. Shao, J. Ding, D. Jin, and Y. Li, "Spatio-temporal few-shot learning via diffusive neural network generation," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=QyFm3D3Tzi

[8] Y. Yuan, J. Ding, H. Wang, D. Jin, and Y. Li, "Activity trajectory generation via modeling spatiotemporal dynamics," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4752–4762.

[9] Y. Yuan, J. Ding, H. Wang, and D. Jin, "Generating daily activities with need dynamics," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 2, pp. 1–28, 2024.

[10] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 804–17 815, 2020.

[11] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[12] L. Liu, R. Zhang, J. Peng, G. Li, B. Du, and L. Lin, "Attentive crowd flow machines," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1553–1561.

[13] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1720–1730.

[14] Z. Zhou, K. Yang, Y. Liang, B. Wang, H. Chen, and Y. Wang, "Predicting collective human mobility via countering spatiotemporal heterogeneity," *IEEE Transactions on Mobile Computing*, 2023.

[15] R. Li, H. Wang, and Y. Li, "Learning slow and fast system dynamics via automatic separation of time scales," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4380–4390.

[16] Y. Yuan, J. Ding, C. Shao, D. Jin, and Y. Li, "Spatio-temporal diffusion point processes," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 3173–3184.

[17] Z. Wang, Y. Li, X. Chen, S.-N. Lim, A. Torralba, H. Zhao, and S. Wang, "Detecting everything in the open world: Towards universal object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 433–11 443.

[18] Y. Bai, X. Geng, K. Mangalam, A. Bar, A. Yuille, T. Darrell, J. Malik, and A. A. Efros, "Sequential modeling enables scalable learning for large vision models," *arXiv preprint arXiv:2312.00785*, 2023.

[19] A. Garza and M. Mergenthaler-Canseco, "Timegpt-1," *arXiv preprint arXiv:2310.03589*, 2023.

[20] Y. Yuan, J. Ding, J. Feng, D. Jin, and Y. Li, "Unist: a prompt-empowered universal model for urban spatio-temporal prediction," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4095–4106.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[22] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[23] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms," *Advances in neural information processing systems*, vol. 30, 2017.

[24] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention convlstm for spatiotemporal prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 531–11 538.

[25] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 1020–1027.

[26] Z. Zhang, Z. Huang, Z. Hu, X. Zhao, W. Wang, Z. Liu, J. Zhang, S. J. Qin, and H. Zhao, "Mlpst: Mlp is all you need for spatio-temporal prediction," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3381–3390.

[27] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4454–4458.

[28] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3656–3663.

[29] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 1, pp. 1–21, 2023.

[30] C. Chen, Y. Liu, L. Chen, and C. Zhang, "Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

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

[32] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 2020, pp. 507–523.

[33] Z. Zhou, J. Ding, Y. Liu, D. Jin, and Y. Li, "Towards generative modeling of urban flow through knowledge-enhanced denoising diffusion," in *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 2023, pp. 1–12.

[34] B. Lu, X. Gan, W. Zhang, H. Yao, L. Fu, and X. Wang, "Spatio-temporal graph few-shot learning with cross-city knowledge transfer," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1162–1172.

[35] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in *The world wide web conference*, 2019, pp. 2181–2191.

[36] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, Z. Junbo, and Y. Zheng, "Spatio-temporal self-supervised learning for traffic flow prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, pp. 4356–4364, 2023.

[37] X. Zhang, Y. Gong, X. Zhang, X. Wu, C. Zhang, and X. Dong, "Mask-and contrast-enhanced spatio-temporal learning for urban flow prediction," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3298–3307.

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

[39] Y. Tang, A. Qu, A. H. Chow, W. H. Lam, S. Wong, and W. Ma, "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*, 2022, pp. 1905–1915.

[40] Y. Yuan, H. Wang, J. Ding, D. Jin, and Y. Li, "Learning to simulate daily activities via modeling dynamic human needs," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 906–916.

[41] Z. Liu, G. Zheng, and Y. Yu, "Cross-city few-shot traffic forecasting via traffic pattern bank," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 1451–1460.

[42] Z. Zhang, X. Zhao, Q. Liu, C. Zhang, Q. Ma, W. Wang, H. Zhao, Y. Wang, and Z. Liu, "Promptst: Prompt-enhanced spatio-temporal multi-attribute prediction," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3195–3205.

[43] Z. Li, L. Xia, Y. Xu, and C. Huang, "Generative pre-training of spatio-temporal graph neural networks," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=nMH5cUaSj8

[44] Z. Shao, Z. Zhang, F. Wang, and Y. Xu, "Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting," in *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*. ACM, 2022, pp. 1567–1577.

[45] K. Jin, J. Wi, E. Lee, S. Kang, S. Kim, and Y. Kim, "Trafficbert: Pre-trained model with large-scale data for long-range traffic flow forecasting," *Expert Systems with Applications*, vol. 186, p. 115738, 2021.

[46] X. Wang, D. Wang, L. Chen, and Y. Lin, "Building transportation foundation model via generative graph transformer," 2023.

[47] Z. Li, L. Xia, J. Tang, Y. Xu, L. Shi, L. Xia, D. Yin, and C. Huang, "Urbangpt: Spatio-temporal large language models," 2024.

[48] L. Liu, S. Yu, R. Wang, Z. Ma, and Y. Shen, "How can large language models understand spatial-temporal data?" 2024.

[49] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[50] J. Feng, Y. Du, T. Liu, S. Guo, Y. Lin, and Y. Li, "Citygpt: Empowering urban spatial cognition of large language models," *arXiv preprint arXiv:2406.13948*, 2024.

[51] F. Xu, J. Zhang, C. Gao, J. Feng, and Y. Li, "Urban generative intelligence (ugi): A foundational platform for agents in embodied city environment," *arXiv preprint arXiv:2312.11813*, 2023.

[52] J. Feng, J. Zhang, J. Yan, X. Zhang, T. Ouyang, T. Liu, Y. Du, S. Guo, and Y. Li, "Citybench: Evaluating the capabilities of large language model as world model," *arXiv preprint arXiv:2406.13945*, 2024.

[53] Y. Yan, H. Wen, S. Zhong, W. Chen, H. Chen, Q. Wen, R. Zimmermann, and Y. Liang, "When urban region profiling meets large language models," *arXiv preprint arXiv:2310.18340*, 2023.

[54] J. Gong, J. Ding, F. Meng, G. Chen, H. Chen, S. Zhao, H. Lu, and Y. Li, "A population-to-individual tuning framework for adapting pretrained lm to on-device user intent prediction," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3637528.3671984

[55] S. Lai, Z. Xu, W. Zhang, H. Liu, and H. Xiong, "Large language models as traffic signal control agents: Capacity and opportunity," *arXiv preprint arXiv:2312.16044*, 2023.

[56] M. Jin, Q. Wen, Y. Liang, C. Zhang, S. Xue, X. Wang, J. Zhang, Y. Wang, H. Chen, X. Li *et al.*, "Large models for time series and spatio-temporal data: A survey and outlook," *arXiv preprint arXiv:2310.10196*, 2023.

[57] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, "Tempo: Prompt-based generative pre-trained transformer for time series forecasting," *arXiv preprint arXiv:2310.04948*, 2023.

[58] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan *et al.*, "Time-llm: Time series forecasting by reprogramming large language models," *arXiv preprint arXiv:2310.01728*, 2023.

[59] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One fits all: Power general time series analysis by pretrained lm," *arXiv preprint arXiv:2302.11939*, 2023.

[60] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[61] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision*. Springer, 2022, pp. 709–727.

[62] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," *arXiv preprint arXiv:1802.00386*, 2018.

[63] Y. Jin, K. Chen, and Q. Yang, "Transferable graph structure learning for graph-based traffic forecasting across cities," in *Proceedings of the 29th*

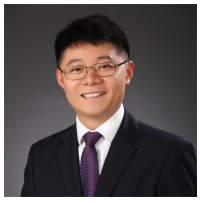
ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 1032–1043.
[64] Z. Fang, D. Wu, L. Pan *et al.*, "When transfer learning meets cross-city urban flow prediction: spatio-temporal adaptation matters," *IJCAI'22*, pp. 2030–2036, 2022.
[65] J. Hu, X. Liu, Z. Fan, Y. Yin, S. Xiang, S. Ramasamy, and R. Zimmermann, "Prompt-based spatio-temporal graph transfer learning," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 890–899.
[66] H. Chai, T. Jiang, and L. Yu, "Diffusion model-based mobile traffic generation with open data for network planning and optimization," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4828–4838.
[67] S. Hui, H. Wang, Z. Wang, X. Yang, Z. Liu, D. Jin, and Y. Li, "Knowledge enhanced gan for iot traffic generation," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3336–3346.
[68] H. Chai, S. Zhang, X. Qi, and Y. Li, "Fomo: A foundation model for mobile traffic forecasting with diffusion model," *arXiv preprint arXiv:2410.15322*, 2024.
[69] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.
[70] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.
[71] Y. Liu, C. Li, J. Wang, and M. Long, "Koopa: Learning non-stationary time series dynamics with koopman predictors," *arXiv preprint arXiv:2305.18803*, 2023.
[72] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," *Advances in neural information processing systems*, vol. 28, 2015.
[73] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.
[74] L. Zhao, M. Gao, and Z. Wang, "St-gsp: Spatial-temporal global semantic representation learning for urban flow prediction," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1443–1451.
[75] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "St-norm: Spatial and temporal normalization for multi-variate time series forecasting," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 269–278.
[76] Z. Chang, X. Zhang, S. Wang, S. Ma, Y. Ye, X. Xinguang, and W. Gao, "Mau: A motion-aware unit for video prediction and beyond," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 950–26 962, 2021.
[77] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu, "Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9154–9162.
[78] Z. Gao, C. Tan, L. Wu, and S. Z. Li, "Simvp: Simpler yet better video prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3170–3180.
[79] C. Tan, Z. Gao, L. Wu, Y. Xu, J. Xia, S. Li, and S. Z. Li, "Temporal attention unit: Towards efficient spatiotemporal predictive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 770–18 782.
[80] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.
[81] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.
[82] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.
[83] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.
[84] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, ser. IJCAI-2018, 2018.

Yuan Yuan received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2020. She is currently working towards the PhD degree with the Department of Electronic Engineering, Tsinghua University (advised by Prof. Yong Li and Prof. Depeng Jin). Her current research interests mainly focus on urban foundation models, urban simulations and generative modeling of spatio-temporal systems. She has publications in conferences and journals such as ICLR, KDD, WWW, and TKDE, etc.

Jingtao Ding received the B.S. degrees in electronic engineering and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2015 and 2020, respectively. He is currently a Post-Doctoral Research Fellow with the Department of Electronic Engineering, Tsinghua University. His research interests include mobile computing, spatiotemporal data mining and user behavior modeling. He has over 30 publications in journals and conferences such as TKDE, TOIS, KDD, NeurIPS, WWW, ICLR, SIGIR, IJCAI, etc.

Jie Feng is currently a postdoctoral researcher at the Department of Electronic Engineering in Tsinghua University. Previously, he worked at Meituan as a researcher specializing in intelligent decision-making and large language models from 2021 to 2023. He received his B.S. and Ph.D. degrees in electrical engineering from Tsinghua University in 2016 and 2021, respectively. His research mainly focuses on spatiotemporal data mining and urban foundation models, with over 20 papers published in top-tier venues including WWW, KDD, UbiComp, AAAI, TKDE, and others. His research is supported by the Shuimu Tsinghua Scholar Program and the China Postdoctoral Talent Plan.

Depeng Jin (M'2009) received his B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1995 and 1999 respectively both in electronics engineering. Now he is an associate professor at Tsinghua University and vice chair of Department of Electronic Engineering. Dr. Jin was awarded National Scientific and Technological Innovation Prize (Second Class) in 2002. His research fields include telecommunications, high-speed networks, ASIC design and future internet architecture.



Yong Li (M'09-SM'16) is currently a full Professor of the Department of Electronic Engineering, Tsinghua University. He received the Ph.D. degree in electronic engineering from Tsinghua University in 2012. His research interests include machine learning and data mining, particularly, automatic machine learning and spatial-temporal data mining for urban computing, recommender systems, and knowledge graphs. Dr. Li has served as General Chair, TPC Chair, SPC/TPC Member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. He has published over 100 papers on first-tier international conferences and journals, including KDD, ICLR, NeurIPS, WWW, UbiComp, SIGIR, AAAI, TKDE, TMC etc, and his papers have total citations more than 27000.

APPENDIX

A. Datasets

1) Basic Information: Here we provide more details of the used datasets in our study. We collect various spatio-temporal data from multiple cities and domains. Table VIII summarizes the basic information of the used datasets, and Table IX reports the basic statistics. Specifically, values for Crowd and Cellular datasets in Table II, Table III, Table XIV, Table XV and Figure 4 should be scaled by a factor of 10^3.

2) Data Preprocessing: For each dataset, We split it into three non-overlapping periods: the first 70% of the period was used as the training set, the next 15% as the validation set, and the final 15% as the test set. To ensure no overlap between train/val/test sets, we removed intermediate sequences. We have normalized all datasets to the range $[-1, 1]$. The reported prediction results are denormalized results.

B. Baselines

- **HA**: History average uses the mean value of historical data for future predictions. Here we use historical data of corresponding periods in the past days.
- **ARIMA**: Auto-regressive Integrated Moving Average model a widely used statistical method for time series forecasting. It is a powerful tool for analyzing and predicting time series data, which are observations collected at regular intervals over time.
- **STResNet** [11]: It is a spatio-temporal model for crowd flow prediction, which utilizes residual neural networks to model the temporal closeness, period, and trend properties.
- **ACFM** [12]: Attentive Crowd Flow Machine model is proposed to predict the dynamics of the crowd flows. It learns the dynamics by leveraging an attention mechanism to adaptively aggregate the sequential patterns and the periodic patterns.
- **STGSP** [74]: This model propose that the global information and positional information in the temporal dimension are important for spatio-temporal prediction. To this end, it leverages a semantic flow encoder to model the temporal relative positional signals. Besides, it utilizes an attention mechanism to capture the multi-scale temporal dependencies.
- **MC-STL** [37]: It leverages an state-of-the-art training techniques for spatio-temporal predition, the mask-enhanced contrastive learning, which can effectively capture the relationships on the spatio-temporal dimension.
- **MAU** [76]: Motion-aware unit is a video prediction model. it broadens the temporal receptive fields of prediction units, which can facilitates to capture inter-frame motion correlations. It consists of an attention module and a fusion module.
- **PredRNN** [23]: PredRNN is a recurrent network-based model. In this model, the memory cells are explicitly decoupled, and they calculate in independent transition manners. Besides, different from the memory cell of LSTM, this network leverages zigzap memory flow, which facilitates to learn at distinct levels.
- **MIM** [77]: Memory utilize the differential information between adjacent recurrent states, which facilitates to model the non-stationary properties. Stacked multiple MIM blocks make it possible to model high-order non-stationarity.
- **SimVP** [78]: It is a simple yet very effective video prediction model. It is completely built based on convolutional neural networks and uses MSE loss. It serves as a solid baseline in video prediction tasks.
- **TAU** [79]: Temporal Attention Unit is the state-of-the-art video prediction model. It decomposes the temporal attention into two parts: intra-frame attention and inter-frame attention, which are static and dynamical, respectively. Besides, it introduces a novel regularization, *i.e.,* differential divergence regularization, to consider the impact of inter-frame variations.
- **STID** [27]: It is a MLP-based spatio-temporal prediction model, which is simple yet effective. Its superior performance comes from the identification of the indistinguishability of samples in spatio-temporal dimensions. It demonstrates that it is promising to design efficient and effective models in spatio-temporal predictions.
- **STNorm** [75]: It proposed two types of normalization modules: spatial normalization and temporal normalization. These two normalization methods can separately consider high-frequency components and local components.
- **PatchTST** [80]: It first employed patching and self-supervised learning in multivariate time series forecasting. It has two essential designs: (i) segmenting the original time series into patches to capture long-term correlations, (ii) different channels are operated independently, which share the same network.
- **iTransformer** [81]: This is the state-of-the-art multivariate time series model. Different from other Transformer-based methods, it employs the attention and feed-forward operation on an inverted dimension, that is, the multivariate correlation.
- **MAML** [82]: Model-Agnostic Meta-Learning is an state-of-the-art meta learning technique. The main idea is to learn a good initialization from various tasks for the target task.
- **MetaST** [35]: It is an urban transfer learning approach, which utilizes long-period data from multiple cities for transfer learning. by employing a meta-learning approach, it learns a generalized network initialization adaptable to target cities. It also incorporates a pattern-based spatial-temporal memory to capture important patterns.
- **PromptST [42]**: It is the state-of-the-art pre-training and prompt-tuning approach for spatio-temporal prediction.

C. Algorithms

We provide the training algorithm for spatio-temporal pretrianing on multiple datasets in Algorithm 1. We also present the prompt fine-tuning algorithm in Algorithm 2.

D. Implementation Details

1) Evaluation Metrics: We use commonly used regression metrics, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), to measure the prediction performance. Suppose $\boldsymbol{Y} = Y_1, ..., Y_M$ are ground truth for real spatio-temporal data, $\hat{\boldsymbol{Y}} = \hat{Y}_1, ..., \hat{Y}_N$ are the predicted values by the model,

TABLE VIII: The basic information of the used datasets.

Dataset	Domain	City	Temporal Duration	Temporal interval	Spatial partition
TaxiBJ	Taxi GPS	Beijing, China	20130601-20131030 20140301-20140630 20150301-20150630 20151101-20160410	Half an hour	32×32
Cellular	Cellular usage	Nanjing, China	20201111-20210531	Half an hour	16 * 20
TaxiNYC-1	Taxi OD	New York City, USA	20160101-20160229	Half an hour	16 * 12
TaxiNYC-2	Taxi OD	New York City, USA	20150101-20150301	Half an hour	20 * 10
BikeNYC-1	Bike usage	New York City, USA	20160801-20160929	One hour	16 * 8
BikeNYC-2	Bike usage	New York City, USA	20160701-20160829	Half an hour	10 * 20
TDrive	Taxi trajectory	New York City, USA	20150201-20160602	One hour	32×32
Crowd	Crowd flow	Nanjing, China	20201111-20210531	Half an hour	16 * 20
TrafficCS	Traffic speed	Changsha, China	20220305-20220405	Five minutes	28×28
TrafficWH	Traffic speed	Wuhan, China	20220305-20220405	Five minutes	30×28
TrafficCD	Traffic speed	Chengdu, China	20220305-20220405	Five minutes	28×26
TrafficJN	Traffic speed	Jinan, China	20220305-20220405	Five minutes	32×18
TrafficNJ	Traffic speed	Nanjing, China	20220305-20220405	Five minutes	32×24
TrafficSH	Traffic speed	Shanghai, China	20220127-20220227	Five minutes	28×32
TrafficSZ	Traffic speed	Shenzhen, China	20220305-20220405	Five minutes	24×18
TrafficGZ	Traffic speed	Guangzhou, China	20220305-20220405	Five minutes	32×26
TrafficGY	Traffic speed	Guiyang, China	20220305-20220405	Five minutes	26×28
TrafficTJ	Traffic speed	Tianjin, China	20220305-20220405	Five minutes	24×30
TrafficHZ	Traffic speed	Hangzhou, China	20220305-20220405	Five minutes	28×24
TrafficZZ	Traffic speed	Zhengzhou, China	20220305-20220405	Five minutes	26×26
TrafficBJ	Traffic speed	Beijing, China	20220305-20220405	Five minutes	30×32

TABLE IX: The basic statistics of the used datasets.

Dataset	Min value	Max value	Mean value	Standard deviation
TaxiBJ	0.0	1285	107	133
Cellular	0.0	2992532	75258	149505
TaxiNYC-1	0.0	1517	32	94
TaxiNYC-2	0.0	1283	37	102
BikeNYC-1	0.0	266	9.2	18.1
BikeNYC-2	0.0	299	4.4	14.6
TDrive	0.0	2681	123	229
Crowd	0.0	593118	21656	40825
TrafficCS	0.0	22.25	6.22	4.79
TrafficWH	0.0	22.35	6.22	4.68
TrafficCD	0.0	22.25	7.33	4.36
TrafficJN	0.0	25.04	5.72	4.71
TrafficNJ	0.0	24.82	5.38	4.73
TrafficSH	0.0	21.83	7.92	3.86
TrafficSZ	0.0	22.12	5.11	4.75
TrafficGZ	0.0	25.16	5.26	4.79
TrafficGY	0.0	28.89	5.95	7.03
TrafficTJ	0.0	25.24	6.32	5.05
TrafficHZ	0.0	29.50	3.81	4.38
TrafficZZ	0.0	23.26	6.67	4.32
TrafficBJ	0.0	22.82	6.30	4.22

and N is the number of total testing samples, These two metrics can be formulated as follows:

$$\begin{aligned} \mathrm{RMSE}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) &= \sqrt{\frac{1}{N}\sum_{i}^{N}\left(Y_i - \hat{Y}_i\right)^2}, \\ \mathrm{MAE}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) &= \frac{1}{N}\sum_{i}^{N}\left|Y_i - \hat{Y}_i\right|, \end{aligned} \tag{1}$$

2) Prompt-Tuning: The prompt-tuning stage aims to train a effective prompt network, which generates customized prompt for specific spatio-temporal pattern. We propose to leverage four types of spatio-temporal knowledge: (i) spatial closess (s_c), (ii) spatial hierarchy (s_h), (iii) temporal closeness (t_c), and (iv) temporal period (t_p). These knowledge-guided features are extracted from the input sequence. The input is the historical spatio-temporal sequence, the output is the predicted future spatio-temporal sequence, and the objective is to minimize the distance between the predicted results and real data. Specifically, we use the widely adopted mean squared error loss function with l_2 regularization on the parameters in UniST to prevent over-fitting, which can be formulated as follows

TABLE X: The parameter details of UniST with different sizes evaluated in ablation studies.

Model	#Encoder Layers	#Decoder Layers	Hidden Dimension (Encoder)	Decoder Hidden Dimension (Decoder)
2M Params	2	2	64	64
8M Params	4	3	128	128
10M Params	6	4	128	128
15M Params	8	8	128	128
30M Params	6	6	256	256

Algorithm 1 Spatio-temporal Pre-training

Input: Dataset $D = \{D_1, D_2, \ldots, D_M\}$, base spatio-temporal prediction model F, and loss function L.
Initialize: Learnable parameters θ for the model F.
for $epoch \in \{1, 2, \ldots, N_{iter}\}$ **do**
  Randomly sample a dataset D_m and a mini-batch X from D_m.
  Randomly choose a masking strategy M from the four strategies.
  Mask the input X into X_m
  Compute the reconstructions $\hat{y} \leftarrow F_\theta(X_m)$
  Compute the MSE loss $\mathcal{L} \leftarrow L(\hat{y}, y)$
  Update the model's parameters $\theta \leftarrow update(\mathcal{L}; \theta)$
end for=0

Algorithm 2 Prompt Tuning

Input: Dataset $D = \{D_1, D_2, \ldots, D_M\}$, parameters of pre-trained base model θ, and loss function L
Initialize: Learnable parameters ϕ for the prompt network G.
Load the pretrained model parameters θ.
Fix the parameters of the attention and feed-forward layers of the base model F_θ.
for $epoch \in \{1, 2, \ldots, N_{iter}\}$ **do**
  Randomly sample a dataset D_m and a mini-batch (X, Y) from D_m.
  Generate the prompt P for the mini-batch $P \leftarrow G\phi(X)$.
  Add the prompt to the input space $X_p = X + P$.
  Compute the predictions $\hat{y} \leftarrow F_\theta(X_p)$
  Compute the MSE loss $\mathcal{L} \leftarrow L(\hat{y}, Y)$
  Update the model's parameters $\gamma \leftarrow update(\mathcal{L}; \gamma), \theta \leftarrow update(\mathcal{L}; \theta)$
end for=0

$$\mathcal{L} = \frac{1}{M} \sum (\hat{y} - y)^2 + \lambda \sum_{\theta \in \Theta} \|\theta\|_2 \tag{2}$$

where $\hat{y}$ and y are ground truths and model predictions, respectively; Θ denotes the set that contains all model parameters.

3) Baseline Implementation: We compare UniST with a broad collection of state-of-the-art models for spatio-temporal prediction, which can be categorized into five groups as introduced in Section IV-A. If we consider the scalability to diverse data formats, i.e., different spatio-temporal data shapes, these baselines can be categorized into two groups: (i) approaches that are scalable with different spatio-temporal scales, such as PatchTST, MAML, and MetaST, and (ii) approaches that are non-scalable, including deep urban prediction approaches, video prediction approaches, and iTransformer. Most baselines are not scalable to different data shapes because they require a fixed number of spatial grids or variables, as seen in CNN-based approaches, MLP-based approaches, and multivariate time series models. Due to the varied data shapes, non-scalable baselines cannot be trained using all datasets, so we train separate models for each dataset.

For the scalable baseline, PatchTST [80], it utilizes a channel-independent patch time series Transformer architecture, allowing it to be applied to datasets with varied spatio-temporal shapes. To ensure a fair comparison, we train both separate models and a single "one-for-all" model, as shown in Table II.

Notably, there are two baselines employ pretraining and finetuning: PatchTST [80] and PromptST [42]. However, PromptST requires a fixed number of nodes, limiting its flexibility across different data formats. In contrast, the channel-independence of PatchTST allows it to handle varied data shapes. While PromptST is a state-of-the-art pre-training and prompt-tuning approach, it lacks generalization ability across different datasets.

4) Experimental Design: In our experimental design, we incorporate four distinct prediction tasks: short-term prediction, long-term prediction, few-shot prediction, and zero-shot prediction. This design aligns with established practices in foundation models for time series forecasting [58], [59], [80]. The short-term and long-term prediction tasks are conducted without transfer learning settings. In these tasks, the model is trained on a set of N datasets and then evaluated on the corresponding testing sets from these datasets. This setup enables us to directly assess the model's performance across multiple datasets using a single universal model.

Furthermore, the few-shot and zero-shot prediction tasks are designed to evaluate the model's generalization capabilities. In these tasks, the model learns from a set of source datasets to build a pretrained model and a memory pool, which is then utilized for prediction on target datasets. The key difference between the few-shot and zero-shot settings lies in the fine-tuning process on the target dataset. In few-shot prediction, the model undergoes a limited fine-tuning process using a small percentage of the target dataset's training data, while in zero-shot prediction, the model directly applies the pre-trained model and memory pool to make predictions on the target dataset without any fine-tuning.

These four tasks collectively offer a comprehensive evaluation of the model's performance and its ability to generalize across diverse spatio-temporal datasets.

TABLE XI: Comparison of computational cost and memory cost between different approaches. The training time denotes the time cost to train all instances with one epoch.

Model	STResNet	ACFM	STID	STNorm	STGSP	MC-STL	MAU	PredRNN	MIM	SimVP	TAU	PatchTST	iTransformer	UniST
Model Size (M)	2.51	1.90	1.63	1.15	5.51	6.35	10.55	17.07	26.24	9.96	9.55	2.59	25.27	6.71
Memory Cost (MB)	1475	1671	1715	2539	1459	1607	1579	1065	1241	1039	1075	2859	2935	2875
Training Time (min)	0.057	0.561	0.054	0.461	0.078	0.311	0.828	0.455	0.836	0.224	0.224	0.338	0.093	1.4 (20+ datasets)
Total Training Time (hour)	∼6	∼6	∼14	∼16	∼24	∼22	∼30	∼40	∼50	∼30	∼30	∼15	∼16	∼12
Inference Time (min)	0.011	0.026	0.007	0.070	0.006	0.013	0.026	0.015	0.024	0.013	0.010	0.031	0.012	0.034

Note: The total training time represents the overall training duration for all datasets, and is an estimate denoted by the symbol∼.

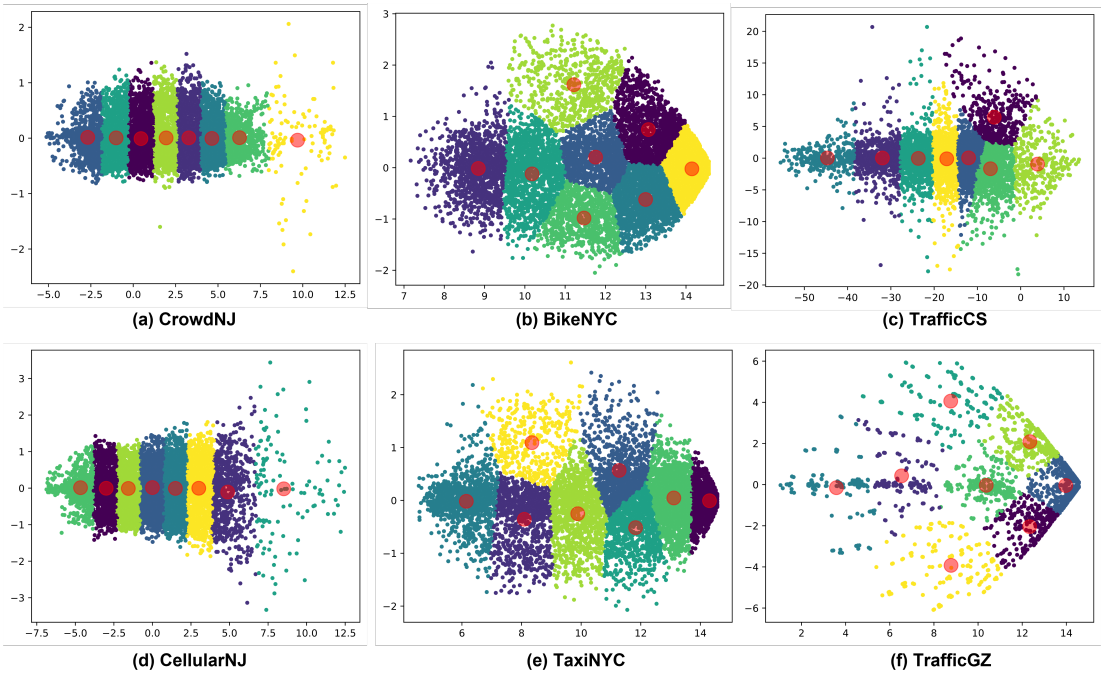


Fig. 13: Visualization of different spatio-temporal datasets: Firstly, the high-dimensional data is reduced to a two-dimensional vector using t-SNE. Subsequently, the embeddings are visualized in clusters using the k-means clustering method.

E. Additional Results

1) Model Efficiency: Table XI shows a detailed comparison of the computational and memory costs of UniST against baselines. The results show that the model size and memory cost of UniST are comparable to those of other approaches. Although UniST's per-epoch training time is longer due to the multi-dataset pre-training process, its overall total training time remains efficient. Notably, UniST consistently outperforms the baselines across all datasets with just a single model, achieving an excellent balance between efficiency and performance.

2) Dataset Similarity: To assess the similarities among the datasets used in our study, we employed a two-step process. First, we reduced the dimension of the spatio-temporal data using t-SNE, a technique for dimension reduction. This allowed us to visualize the datasets in a lower-dimensional space. Second, we applied the k-means clustering method to the reduced data to identify clusters of similar spatio-temporal patterns.

The results of our visualization in Figure 13 revealed interesting insights. We found that certain datasets, such as the Crowd data and Cellular data in Nanjing, exhibited similar spatio-temporal patterns. Similarly, the Bike data and Taxi data in New York City showed similarities in their patterns. However, most datasets from different cities or domains exhibited distinct spatio-temporal patterns, indicating significant distribution shifts. These observations highlight the powerful generalization ability and universality of our approach across datasets with significantly distinct spatio-temporal patterns.

3) Additional Prediction Results: Table XII∼Table XVII report addition prediction results.

TABLE XII: Performance comparison of short-term prediction on seven datasets in terms of MAE and RMSE. We use the average prediction errors over all prediction steps.

	TaxiNYC-1		**BikeNYC-2**		**TaxiNYC-2**		**TrafficBJ**		**TrafficNJ**		**TrafficWH**		**TrafficSZ**	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
HA	57.07	18.57	15.68	7.17	52.84	15.74	1.033	0.582	1.593	0.774	1.351	0.645	0.791	0.416
ARIMA	55.39	20.94	25.01	13.63	62.9	29.56	1.32	0.735	1.30	0.709	1.51	0.748	0.821	0.445
STResNet	29.45	17.96	7.18	3.94	22.16	12.06	0.828	0.547	1.03	0.635	0.903	0.568	0.709	0.465
ACFM	23.35	11.54	5.99	3.094	14.48	6.39	0.706	0.44	0.888	0.515	0.784	0.471	0.573	0.35
STID	17.75	7.03	5.70	2.711	17.37	7.35	0.724	0.431	0.847	0.459	0.78	0.436	0.576	0.33
STNorm	21.26	8.14	6.47	3.03	19.02	7.17	0.727	0.428	0.904	0.476	0.81	0.445	0.666	0.369
STGSP	28.13	10.29	14.20	7.38	29.10	10.14	0.736	0.444	0.883	0.491	0.804	0.473	0.86	0.52
MC-STL	18.44	9.51	6.26	3.40	16.78	8.50	0.975	0.709	1.13	0.78	1.1	0.773	0.83	0.615
MAU	28.70	11.23	6.12	2.95	19.38	7.27	1.12	0.797	0.978	0.545	1.37	0.917	0.826	0.523
PredRNN	16.53	5.80	6.47	3.08	19.89	7.23	0.651	0.376	0.852	0.457	0.74	0.421	0.58	0.335
MIM	18.83	6.866	6.36	2.89	18.02	6.56	2.62	2.14	4.65	3.39	3.86	3.15	2.22	1.40
SimVP	16.63	7.51	5.96	2.92	15.10	6.54	0.664	0.408	0.861	0.481	0.779	0.475	0.583	0.359
TAU	16.91	6.85	5.98	2.89	15.35	6.80	0.70	0.44	0.89	0.528	0.747	0.444	0.576	0.353
PatchTST	41.34	13.10	12.33	5.30	37.76	11.13	0.935	0.512	1.379	0.658	1.17	0.561	0.718	0.370
iTransformer	36.73	13.11	9.86	4.50	33.03	11.22	0.876	0.490	1.18	0.60	1.10	0.542	0.718	0.378
PatchTST(one-for-all)	44.43	14.56	13.62	6.03	41.04	12.61	0.964	0.524	1.42	0.675	1.22	0.581	0.739	0.375
UniST (ours)	15.32	5.65	5.50	2.56	12.71	4.82	0.689	0.387	0.845	0.421	0.762	0.396	0.513	0.264

TABLE XIII: Performance comparison of short-term prediction on seven datasets in terms of MAE and RMSE. We use the average prediction errors over all prediction steps.

	TrafficTJ		**TrafficGY**		**TrafficGZ**		**TrafficZZ**		**TrafficCS**		**TrafficCD**		**TrafficHZ**	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
HA	1.61	0.824	1.79	0.726	0.996	0.52	1.47	0.857	1.31	0.676	1.12	0.668	0.765	0.342
ARIMA	2.02	1.59	1.91	1.16	1.37	0.76	1.78	0.998	1.66	0.923	1.54	0.907	0.803	0.364
STResNet	1.12	0.714	1.32	0.799	0.796	0.515	1.03	0.693	0.986	0.651	0.867	0.576	0.669	0.406
ACFM	0.959	0.574	1.10	0.599	0.701	0.418	0.839	0.526	0.842	0.529	0.757	0.493	0.575	0.316
STID	0.976	0.549	1.04	0.544	0.665	0.362	0.838	0.502	0.855	0.5	0.715	0.44	0.546	0.282
STNorm	0.973	0.533	1.12	0.508	0.693	0.373	0.885	0.538	0.91	0.511	0.786	0.489	0.556	0.260
STGSP	0.989	0.572	1.09	0.649	0.733	0.419	0.831	0.505	0.978	0.587	0.776	0.497	0.616	0.331
MC-STL	1.22	0.856	1.82	1.36	1.04	0.775	1.14	0.81	1.14	0.819	1.00	0.733	0.842	0.606
MAU	0.988	0.549	1.14	0.595	0.74	0.415	1.42	0.934	1.31	0.791	1.25	0.919	0.743	0.377
PredRNN	0.971	0.53	1.16	0.608	0.71	0.42	0.853	0.508	0.909	0.572	0.815	0.513	0.602	0.288
MIM	3.44	2.51	5.68	4.53	3.43	2.80	2.05	1.56	3.57	2.71	2.75	2.26	1.92	1.23
SimVP	1.00	0.597	1.13	0.632	0.667	0.399	0.838	0.526	0.835	0.507	0.775	0.495	0.549	0.301
TAU	1.01	0.606	1.11	0.604	0.65	0.378	0.839	0.527	0.869	0.543	0.768	0.495	0.539	0.289
PatchTST	1.44	0.722	1.58	0.634	0.894	0.448	1.31	0.742	1.18	0.599	1.00	0.577	0.696	0.305
iTransformer	1.26	0.675	1.39	0.621	0.846	0.428	1.19	0.696	1.09	0.572	0.941	0.541	0.66	0.30
PatchTST(one-for-all)	1.49	0.740	1.66	0.684	0.931	0.469	1.35	0.752	1.23	0.620	1.04	0.602	0.726	0.325
UniST (ours)	0.958	0.510	1.03	0.458	0.648	0.325	0.832	0.482	0.791	0.423	0.711	0.415	0.530	0.236

TABLE XIV: Performance comparison of long-term prediction on four datasets in terms of MAE and RMSE. We use the average prediction errors over all prediction steps.

	TaxiBJ		**Cellular**		**BikeNYC-2**		**TDrive**	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
HA	74.07	43.79	77.29	31.89	15.84	7.97	144.65	72.48
ARIMA	100.76	56.04	83.66	35.96	15.29	7.25	270.05	140.80
STResNet	51.36	36.08	33.87	20.87	12.73	7.16	163.88	112.27
ACFM	35.49	22.46	26.40	13.24	13.00	7.09	88.76	42.19
STID	36.98	23.19	22.98	11.71	12.75	8.37	83.70	37.66
STNorm	33.78	19.89	71.05	32.14	12.16	5.99	100.43	49.50
STGSP	70.31	42.76	67.07	31.16	14.50	7.66	83.70	37.26
MC-STL	38.23	26.86	39.74	27.04	12.72	7.96	100.55	59.18
MAU	85.58	60.61	75.84	32.78	12.42	5.82	137.17	76.17
PredRNN	43.89	27.42	46.68	24.96	9.72	4.37	175.32	104.79
MIM	38.10	25.82	79.20	39.27	10.02	4.60	107.06	43.67
SimVP	33.53	19.28	23.84	12.90	10.89	5.51	91.13	39.46
TAU	34.88	19.94	23.00	12.72	11.53	6.11	91.54	41.96
PatchTST	30.64	17.49	23.39	12.42	11.13	5.07	92.03	38.89
PatchTST(one-for-all)	31.58	18.67	27.94	10.89	10.71	4.74	111.56	50.57
iTransformer	32.89	18.60	29.329	11.963	11.54	5.19	93.87	40.16
UniST (ours)	30.46	17.95	20.64	10.43	11.91	5.06	90.60	37.01

TABLE XV: Performance comparison in few-shot and zero-shot (only UniST) settings on the Crowd dataset in terms of MAE and RMSE. 1% , 5%, and 10% denote that only the percentage of training data is utilized. We use the average prediction errors over all prediction steps.

	10%		**5%**		**1%**	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
ATFM	19.842	11.446	19.923	11.687	21.166	12.643
STNorm	14.668	7.050	14.884	7.723	35.959	29.585
STID	14.676	7.280	14.975	8.671	25.905	19.610
PredRNN	19.604	9.668	20.186	10.190	24.901	13.142
SimVP	14.093	7.101	14.167	8.550	14.252	8.776
TAU	14.229	7.140	14.456	8.411	14.919	9.096
MAML	14.089	7.180	14.795	8.154	14.334	8.608
MetaST	13.801	6.847	14.220	7.442	14.242	7.949
PatchTST	14.060	6.787	14.142	6.811	14.491	7.227
UniST (few-shot)	13.411	6.365	13.859	6.542	13.952	6.581
UniST (zero-shot)	14.665	7.051	14.665	7.051	14.665	7.051

TABLE XVI: Performance comparison in few-shot and zero-shot (only UniST) settings on the BikeNYC dataset in terms of MAE and RMSE. 1% , 5%, and 10% denote that only the percentage of training data is utilized. We use the average prediction errors over all prediction steps.

	10%		**5%**		**1%**	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
ATFM	8.026	3.511	10.438	4.582	11.876	5.990
STNorm	7.42	2.70	10.21	4.17	12.94	5.20
STID	6.97	3.49	12.46	7.56	15.08	9.38
PredRNN	11.05	4.00	11.29	4.46	12.58	4.75
SimVP	6.570	2.691	8.525	3.174	8.661	3.721
TAU	7.06	3.07	8.74	3.28	8.50	3.72
MAML	6.49	2.31	8.89	3.68	8.98	3.91
MetaST	6.21	2.18	8.22	3.03	8.58	3.60
PatchTST	9.14	2.68	10.09	2.88	9.74	3.86
UniST	5.318	1.668	6.113	1.964	7.811	2.72
UniST (zero-shot)	9.06	3.63	9.06	3.63	9.06	3.63

TABLE XVII: Performance comparison in few-shot and zero-shot (only UniST) settings on the TaxiBJ dataset in terms of MAE and RMSE. 1% , 5%, and 10% denote that only the percentage of training data is utilized. We use the average prediction errors over all prediction steps.

	10%		**5%**		**1%**	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
ATFM	50.631	33.035	55.770	39.205	64.590	44.928
STNorm	39.35	22.48	42.67	26.78	44.76	28.24
STID	34.53	20.54	37.39	24.35	47.94	31.94
PredRNN	84.28	58.52	97.74	73.40	92.21	66.76
SimVP	35.114	20.87	37.42	23.131	40.465	24.95
TAU	37.70	22.69	39.77	25.73	41.98	26.48
MAML	36.24	20.91	36.12	23.47	40.11	24.79
MetaST	35.42	18.65	35.21	21.74	39.08	23.88
PatchTST	44.03	22.69	44.24	22.62	46.43	24.77
UniST	27.59	15.18	31.19	17.58	35.09	20.62
UniST (zero-shot)	51.4	33.1	51.4	33.1	51.4	33.1