Modeling Spatio-Temporal Mobility across Data Silos via Personalized Federated Learning

Yudong Zhang, *Student Member, IEEE,* Xu Wang✉, Pengkun Wang, *Member, IEEE,* Binwu Wang, Zhengyang Zhou, *Member, IEEE,* and Yang Wang✉, *Senior Member, IEEE*

Abstract—Spatio-temporal mobility modeling plays a pivotal role in the advancement of mobile computing. Nowadays, data is frequently held by various distributed silos, which are isolated from each other and confront limitations on data sharing. Given this, there have been some attempts to introduce federated learning into spatio-temporal mobility modeling. Meanwhile, the distributional heterogeneity inherent in the spatio-temporal data also puts forward requirements for model personalization. However, the existing methods tackle personalization in a model-centric manner and fail to explore the data characteristics in various data silos, thus ignoring the fact that the fundamental cause of insufficient personalization in the model is the heterogeneous distribution of data. In this paper, we propose a novel distribution-oriented personalized Federated learning framework for Cross-silo Spatio-Temporal mobility modeling (named **FedCroST**), that leverages learnable spatio-temporal prompts to implicitly represent the local data distribution patterns of data silos and guide the local models to learn the personalized information. Specifically, we focus on the potential characteristics within temporal distribution and devise a conditional diffusion module to generate temporal prompts that serve as guidance for the evolution of the time series. Simultaneously, we emphasize the structure distribution inherent in node neighborhoods and propose adaptive spatial structure partition to construct the spatial prompts, augmenting the spatial information representation. Furthermore, we introduce a denoising autoencoder to effectively harness the learned multi-view spatio-temporal features and obtain personalized representations adapted to local tasks. Our proposal highlights the significance of latent spatio-temporal data distributions in enabling personalized federated spatio-temporal learning, providing new insights into modeling spatio-temporal mobility in data silo scenarios. Extensive experiments conducted on real-world datasets demonstrate that FedCroST outperforms the advanced baselines by a large margin in diverse cross-silo spatio-temporal mobility modeling tasks.

Index Terms—Spatio-temporal mobility modeling, Data silos, Personalized federated learning, Distribution heterogeneity.

1 Introduction

Spatio-temporal mobility modeling holds a pivotal role in the advancement of mobile computing that focuses on the analysis, representation, and prediction of the evolution patterns of physical entities in both spatial and temporal dimensions [1], [2], [3], [4], and thus providing insights into developing intelligent and location-based mobile systems [5], [6], [7], [8]. This field has gained prominence and contributes to mobile computing services through data-driven predictive analytics across various economic and social domains [9], [10], including but not limited to environment monitoring [11], epidemic control [12], and traffic management [13], [14]. For example, in air quality monitoring, such models enable accurate tracking of pollutant dispersion and identification of pollution sources by analyzing the movement of monitoring devices and the spatial distribution of air quality data [15]. This facilitates timely and precise predictions of air quality levels and supports dynamic adjustments to sensor layouts and environmental warnings. In the realm of health and epidemiology, spatio-temporal mobility modeling aids in tracking disease spread, analyzing healthcare mobility patterns, and predicting outbreaks, which is indispensable for addressing health challenges and supporting timely interventions [16]. Similarly, in traffic management, spatio-temporal models are crucial for understanding traffic flow and congestion patterns. They enable real-time traffic signal adjustments, route optimization, and overall improvement of transportation network efficiency by considering both temporal variations in traffic density and the spatial layout of road networks [17]. Therefore, by capturing both temporal and spatial dimensions of mobile entities, these models provide an essential and comprehensive understanding of the current and future states of the mobile system, making them necessary for effective monitoring and management in urban environments. In recent years, researchers have witnessed a blossom of data-driven solutions for spatio-temporal mobility modeling, with deep learning-based approaches becoming mainstream [18], [19].

As an advanced technique of deep learning, Graph Neural Networks (GNNs) are introduced to spatio-temporal mobility modeling and have achieved notable success underpinned by their inherent ability to discern complex relationships and dependencies within graph-structured data [20], [21], [22], [23]. In particular, STGCN [20] leverages ChebNet graph convolution and 1D convolution to extract spatial dependencies and temporal correlations in traffic data. Graph

- *Yudong Zhang, Xu Wang, Pengkun Wang, Binwu Wang, Zhengyang Zhou, and Yang Wang are with the University of Science and Technology of China, Hefei 230026, China (e-mail: {zyd2020, wbw1995}@mail.ustc.edu.cn, {wx309, pengkun, zzy0929, angyan}@ustc.edu.cn).*
- *✉ Prof. Yang Wang and Dr. Xu Wang are the corresponding authors.*

Manuscript received xx xx, xxxx; revised xx xx, xxxx.
This paper is partially supported by the National Natural Science Foundation of China (No.62072427, No.12227901), the Project of Stable Support for Youth Team in Basic Research Field, Chinese Academy of Sciences (CAS) (No.YSBR-005), and the Academic Leaders Cultivation Program, University of Science and Technology of China (USTC).

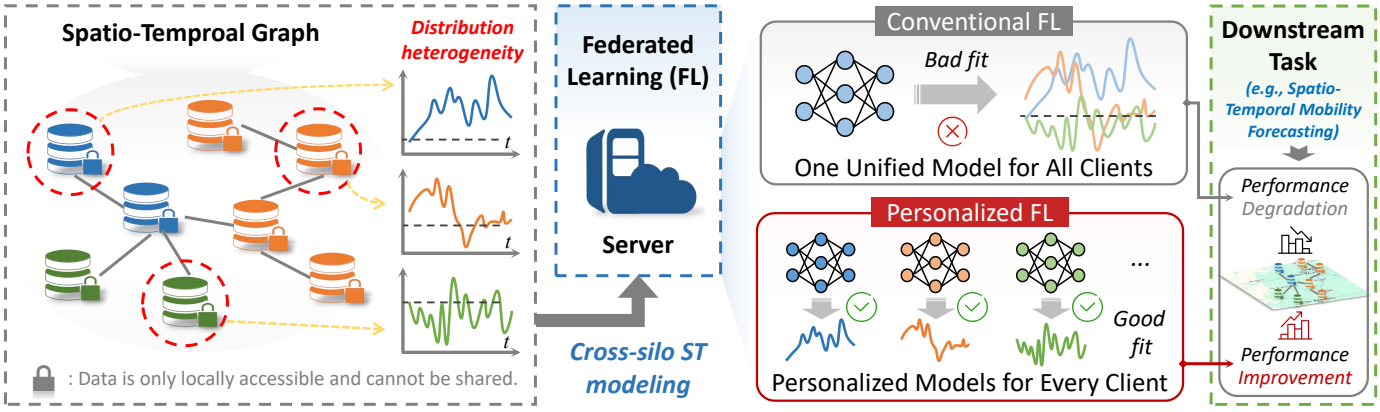

Fig. 1. The basic pipeline of spatio-temporal mobility modeling in data silo scenarios, where federated learning has emerged as the mainstream technology route. Due to the distribution heterogeneity of spatio-temporal data, personalized federated learning has gained increasing attention from researchers, leading to notable improvements in the performance of federated spatio-temporal systems.

WaveNet [24] employs a diffusion convolution layer to capture spatial correlation and utilizes a generic temporal convolution to acquire insights into temporal correlations. DMSTG [17] fuses features from multiple views into their forecasting framework to capture complex spatial-temporal dependencies. Notwithstanding the superiority exhibited by the aforementioned methods, it is imperative to note their common reliance on large-scale datasets centrally collected, which inevitably puts forward strict requirements for the centralized management of data [25]. In the contemporary landscape characterized by an escalating emphasis on data protection, data is frequently held by various distributed entities (e.g., companies or organizations), which are isolated from each other and confront limitations on data sharing [26], [27], [28]. The data separately maintained by each entity is typically insufficient for effective spatio-temporal mobility modeling thus hindering the development of downstream applications. Consequently, there arises a pressing need to delve into the techniques facilitating cross-silo spatio-temporal mobility modeling, aligning with the imperatives of the burgeoning mobile big data era.

To address the above issues, there have been some attempts to introduce distributed learning architecture into spatio-temporal mobility modeling. Federated Learning (FL), a novel distributed computing architecture, has emerged as the predominant paradigm for achieving cross-silo spatio-temporal learning without sharing data. The basic pipeline of introducing federated learning to facilitate spatio-temporal mobility modeling across data silos is illustrated in Fig. 1. In particular, FedGRU [25] integrates emerging FL with a GRU network for spatio-temporal learning, which updates universal learning models through a parameter aggregation mechanism rather than directly sharing raw data among organizations. Recognizing the graph structure inherent in spatio-temporal data, FASTGNN [26] amalgamates a GNN-based model for local training and a novel FL strategy to protect the shared topological information. FedSTN [29] proposes a federated deep learning based on the spatial-temporal long and short-term networks to predict traffic flow by utilizing observed historical traffic data. CNFGNN [30] aggregates local temporal embeddings uploaded from clients and employs GNNs to obtain spatial embeddings, which are sent back to the corresponding clients for forecasting. Nevertheless, as depicted in Fig. 1, the distribution pattern of spatio-temporal data across different data silos is inconsistent, presenting a pervasive challenge termed "distribution heterogeneity" in the field of spatio-temporal mobility modeling [1], [18]. Consequently, the conventional federated learning paradigm, as employed in the aforementioned approaches to construct a unified global model for all clients, proves ineffective in addressing heterogeneous spatio-temporal patterns present in each client, which severely undermines the model's performance in local tasks. The problem is turned around by the proposal of personalized federated learning, which aims to furnish each client with a dedicated model tailored to better fit its local data. For instance, GOF-TTE [31] proposes an online personalized federated learning framework to fill the gap between personal privacy and model performance by dynamically updating the global traffic state. Structured federated learning (SFL) framework [32] learns both the global and personalized models simultaneously using client-wise relation graphs and clients' private data. While these approaches draw inspiration from the generic ideas of personalized federated learning (e.g., regularizing model parameters [33], [34] or adding extra personalized layers for the client-side model [35]), they fall short in contributing to personalization from the perspective of the inherent spatio-temporal nature of the data, specifically addressing the distribution heterogeneity depicted in Fig. 1. As elucidated earlier, the challenge of federated learning in handling spatio-temporal data is fundamentally caused by distribution heterogeneity. Therefore, extracting heterogeneous spatio-temporal patterns among clients from the vantage point of spatio-temporal distribution provides a novel and insightful avenue for enhancing spatio-temporal mobility modeling across silos.

As mentioned earlier, most existing spatio-temporal mobility modeling methods rely on centralized training with large-scale datasets centrally collected, which imposes strict requirements for centralized data management and is not suitable for cross-silo scenarios [25]. While there have been some initial attempts to apply federated learning to spatio-temporal mobility modeling, the inherent distributional het-

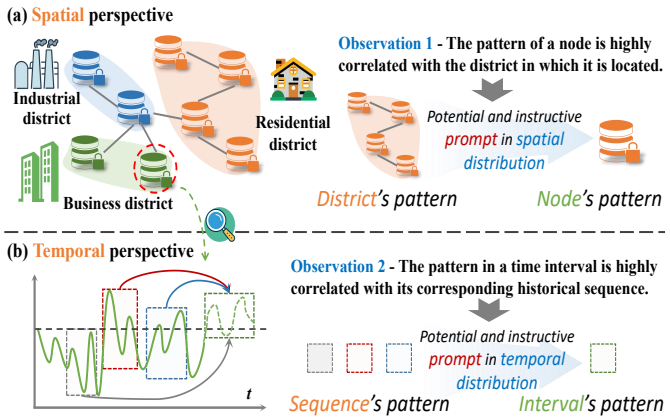

Fig. 2. Spatio-temporal heterogeneity distribution. (a) From a spatial perspective, the pattern of a node is highly correlated with the district in which it is located, i.e., spatial prompt. (b) From a temporal perspective, the pattern in a time interval is highly correlated with its corresponding historical sequence, i.e., temporal prompt.

erogeneity of spatio-temporal data often prevents models from adapting to local data distributions, leading to suboptimal performance in downstream tasks. Therefore, *effectively addressing the problem of spatio-temporal distribution heterogeneity and designing personalized learning architectures adapted to local data distributions remain a significant challenge in cross-silo spatio-temporal mobility modeling*. We systematically deconstruct the spatio-temporal heterogeneity distribution, as depicted in Fig. 2, to unveil distinctive characteristics inherent in spatio-temporal data across both temporal and spatial dimensions. A detailed analysis reveals the following observations: (i) Fig. 2 (a) illustrates that spatio-temporal data is non-Euclidean graph-structured data, where each client corresponds to a node within the graph. Notably, topological connectivity such as geographic proximity establishes relationships between nodes in space, such that nodes do not exist in isolation. Taking the entity in a city as an example, several neighboring nodes with high connectivity may collectively form a functional district with a unique aggregation pattern. Therefore, such macroscopic subgraph patterns imply a priori guiding significance in revealing the internal node pattern. Here, the spatial distribution information, containing the potential and instructive features of the subgraphs-to-nodes, is denoted as spatial prompts. (ii) Fig. 2 (b) highlights that the state of a given time interval is not solely determined by neighboring time intervals, but is also influenced by the overall sequence pattern. However, uncovering this potential trend in the sequence requires delving into the long-term temporal distribution, which is often hindered by the gradient vanishing problem in conventional time-series modeling approaches. Therefore, these long-period sequence patterns are crucial for discerning evolving patterns of spatio-temporal data over the temporal distribution, and we term the instructive temporal distribution information of the sequences-to-intervals as temporal prompts.

To this end, we propose a Spatio-Temporal (ST) distribution-oriented personalized federated learning framework (termed FedCroST) for downstream cross-silo ST mobility modeling by leveraging the intrinsic properties of ST data. Specifically, to unveil latent characteristics within temporal distribution, we devise a hidden state-based conditional diffusion module to generate temporal prompts that serve as guidance for the evolution of the time series. Simultaneously, to capture the structure distribution inherent in node neighborhoods, we propose an adaptive spatial structure partition mechanism, and accordingly learn spatial prompts that augment the spatial information representation. Furthermore, to effectively harness the learned multi-view ST features, we introduce a denoising autoencoder to capture the ST correlations among features, subsequently generating personalized ST representations specifically tailored for local tasks. In summary, the main **contributions** of this paper are as follows:

- *New problem and insight:* To the best of our knowledge, we are the first to highlight the significance of latent ST data distributions in enabling personalized federated ST learning, which provides new insights into modeling ST mobility in the often-overlooked yet increasingly prevalent data silo scenario.
- *Advisable methodologies:* To uncover potential characteristics within temporal distribution, we devise a hidden state-based conditional diffusion module to generate temporal prompts rich in the evolution within time series. To capture the structure distribution inherent in node neighborhoods, we propose adaptive spatial structure partition, and accordingly learn spatial prompts that augment the spatial information representation. Furthermore, to effectively harness the learned multi-view ST features, we introduce a denoising autoencoder to generate personalized ST representations for local tasks.
- *Compelling empirical results:* Extensive experiments are conducted on real-world datasets and the results show that our FedCroST consistently outperforms all the advanced baseline methods in ST mobility modeling across various scenarios.

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries and formalizes the problem. Section 3 investigates the proposed model in detail, followed by our empirical studies in Section 4. Moreover, Section 5 discusses the limitations and future research directions. Finally, Section 6 briefly reviews the related work, and Section 7 concludes this paper.

2 Preliminary

In this section, we will briefly introduce the definition of spatio-temporal mobility modeling and the optimization object under the federated learning setting. Table 1 presents the frequently used notations and corresponding descriptions throughout this paper.

2.1 Spatio-Temporal Mobility Modeling

Spatio-temporal mobility modeling involves constructing computational models to discover the evolving mobility patterns in time and space of key factors within entities, based on the data of these entities in a mobile network [6], [36]. In the context of data silos, spatio-temporal data is typically held by N distributed entities without data sharing [3], [26]. Notably, there is a spatial relationship of graph

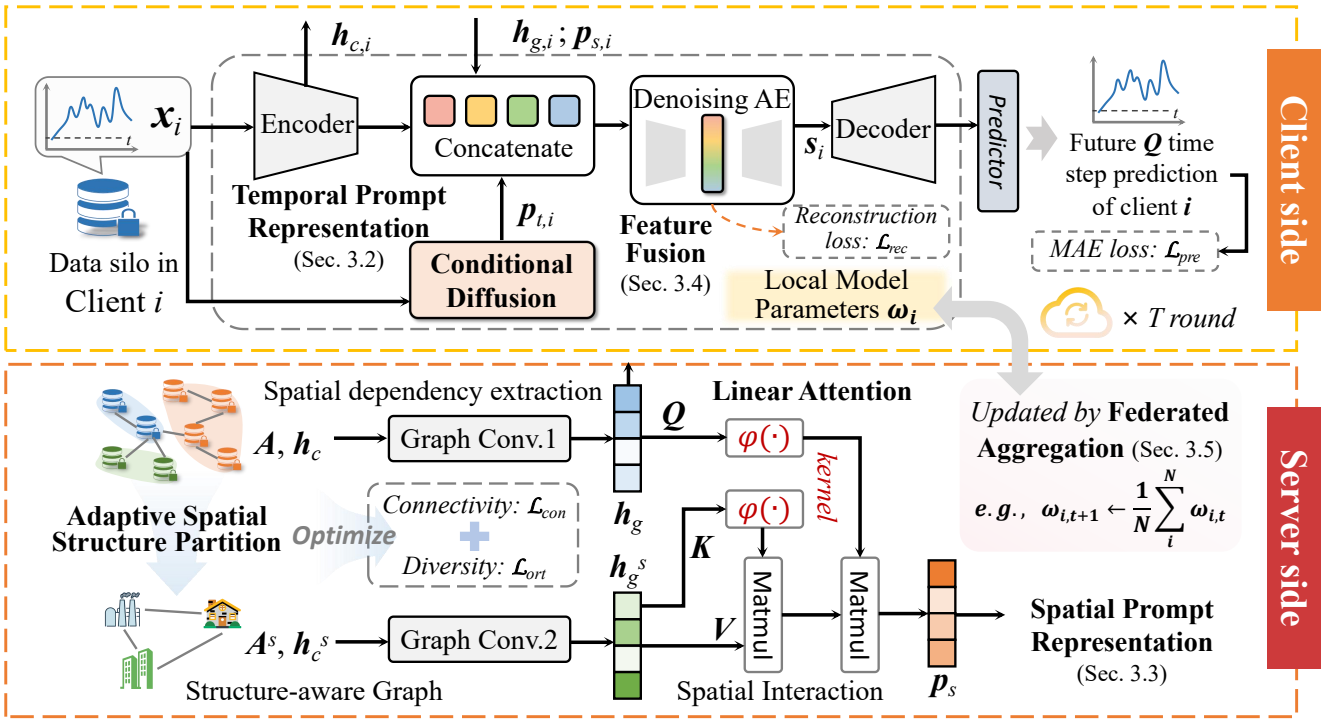

Fig. 3. Overview of FedCroST: A personalized federated learning framework for cross-silo spatio-temporal mobility modeling, following the client-server architecture of the generic FL paradigm. The client-side model includes a temporal prompt representation module and a feature fusion module, while the server focuses on spatial prompt representation and federated parameter aggregation.

TABLE 1
Key Notations and Corresponding Descriptions

Notations	Descriptions
$\boldsymbol{x}_i$	Spatio-temporal mobility data solely held by client i.
$\mathcal{D}_i$	The unshareable local dataset in client i.
$\boldsymbol{A}$, $\boldsymbol{A}^s$	Adjacency matrices of original spatio-temporal graph and structure-aware graph.
N, N^s	Numbers of nodes in original spatio-temporal graph and structure-aware graph.
P, Q	Time steps of historical sequence, forecasting sequence.
ω_i	The parameters of personalized models for client i.
$\mathcal{L}_i$	Loss function of the local model in client i.
$\boldsymbol{h}_t$	The hidden state from the client-side encoder at time step t.
$\beta_1, \cdots, \beta_N$	The variance schedule in diffusion model.
ϵ_θ	The trainable denoising function in the diffusion model.
$\boldsymbol{p}_{t,i}$	The temporal prompts of client i.
$\boldsymbol{S}$	Assignment matrix in adaptive spatial structure partition.
$\boldsymbol{h}_c$, $\boldsymbol{h}_c^s$	The temporal embeddings of original spatio-temporal graph and structure-aware graph.
$\boldsymbol{h}_g$, $\boldsymbol{h}_g^s$	The spatial embeddings of original spatio-temporal graph and structure-aware graph.
$\boldsymbol{p}_{s,i}$	The spatial prompts of client i.
$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$	Query vector, key vector, and value vector in the attention mechanism.
$\boldsymbol{o}_i$	The concatenated feature of client i.
$\boldsymbol{z}_i$	The generated feature in DAE of client i.
$\boldsymbol{g}_i$	The reconstructed feature in DAE of client i.

structure between these entities, with each entity serving as a node in the graph.

Inherited from the existing work [37], [38], [39], [40], we denote the spatial structure of nodes as a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$, where $\mathcal{V}$ is the set of $|\mathcal{V}| = N$ nodes, $\mathcal{E}$ is the set of edges connecting the nodes, and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix representing the proximities among nodes. The spatio-temporal data held by node i at time step t are denoted as $\boldsymbol{x}_{i,t} \in \mathbb{R}^{1 \times D}$, where $t \in \{1, \cdots, P\}$, P is the time steps of historical sequence, and D is the feature dimension.

As a classical spatio-temporal mobility modeling task, predictive analytics is fundamental research in mobile computing [18], [41], [42]. Therefore, this paper focuses on spatio-temporal mobility prediction as a key problem to investigate the modeling of future patterns from the historical spatio-temporal states of entities. The spatio-temporal forecasting task in a specific node i aims to learn a function $\mathcal{F}$ that is able to forecast Q-step future sequences given P-step historical sequences:

$$(\boldsymbol{x}_{i,1}, \cdots, \boldsymbol{x}_{i,t}, \cdots, \boldsymbol{x}_{i,P}) \overset{\mathcal{F}}{\rightarrow} (\boldsymbol{x}_{i,P+1}, \cdots, \boldsymbol{x}_{i,P+Q}), \quad i \in \{1, 2, \cdots, N\}, \tag{1}$$

where $\mathcal{F} : \mathbb{R}^{D \times P} \rightarrow \mathbb{R}^{D \times Q}$.

2.2 Optimization Object in Federated Learning

We denote each data silo as a node, corresponding to a client within the federated system. The datasets across N clients are denoted as $\{\mathcal{D}_1, \cdots, \mathcal{D}_N\}$. Our purpose is to train personalized models ω_i for different clients [43], which is defined as the following optimization problem:

$$\underset{W}{arg\,min} \mathcal{L}(\mathcal{W}) = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \mathcal{L}_i \left(\mathcal{F}_{\omega_i}\right), \tag{2}$$

where $\mathcal{W} = \{\omega_1, \cdots, \omega_N\}$ signifies the parameter set of personalized models for all clients, and $\mathcal{L}_i$ represents the loss function of the local model in client i.

3 METHODOLOGY

3.1 Overview

Fig. 3 provides an overview of FedCroST, which follows the generic FL paradigm of the client-server architecture [30]. Our purpose is to train a personalized model for each client by leveraging the learnable representation of spatiotemporal data distribution. Firstly, we leverage the hidden state of the client-side encoder to generate temporal prompts through a diffusion model. Subsequently, an adaptive spatial structure partition is performed on the original spatio-temporal graph, yielding a structure-aware graph that serves as the basis for generating spatial prompts. The two kinds of prompts can reflect the distinctive characteristics of data distribution in each client from both temporal and spatial perspectives. The generated spatio-temporal prompts play a crucial role in guiding the training process of client-side models, enabling them to adapt to the local data distribution, and facilitating the models to learn personalized information. Detailed explanations of each component will be elaborated in the subsequent sections.

3.2 Temporal Prompt Representation

As Fig. 2 (b) shows, the state of a given time interval is not solely determined by neighboring time intervals, but is also influenced by the overall sequence. We term the instructive temporal distribution information of sequences-to-intervals as "temporal prompts". However, uncovering this potential trend information in the sequence requires delving into the high-dimensional temporal distribution, a challenge often encountered in traditional time-series modeling approaches.

Fortunately, the development of the diffusion model [44] provides a viable solution, which is a prominent technique in generative tasks like image generation and audio synthesis. The key insight of the diffusion model is to approximate the distribution by learning how data can be recovered after it has been diffused to pure noise and attempt to transform a Gaussian distribution back into the data distribution [45], which exactly aligns with our purpose.

3.2.1 Conditional Diffusion Model

Several endeavors have been undertaken to generalize the diffusion model for time series data [46], [47], aiming to extract high-dimensional However, the learning process of vanilla unconditional diffusion models only focuses on the states of neighboring steps, while the distributional representation of time series not only focuses on the neighboring states, but also highly relies on the long-term historical states of the sequence, which makes the vanilla unconditional diffusion models unable to meet the needs of this paper. For this reason, we modify the diffusion model by incorporating hidden states containing the historical information in each step to accurately guide the denoising process, thus approximating the actual distribution to a great extent. As a result, the conditional diffusion model we designed has the ability to effectively enable the generation of temporal prompts to capture the potential correlation between the current state and historical distribution, which well solves the problem of distribution heterogeneity of local data in the time dimension for downstream tasks.

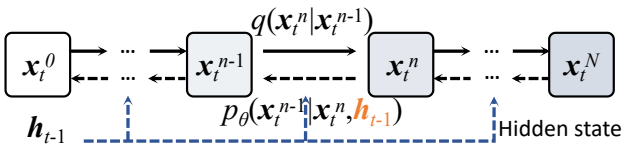

Fig. 4. Detailed process of the conditional diffusion model. Distinguishing from vanilla diffusion models, the proposed method incorporates hidden states as conditional inputs in the denoising process to guide the learning of potential distributions within the time series.

Fig. 4 illustrates the detailed execution process of the conditional diffusion model, encompassing both forward and reverse phases. Different from the vanilla unconditional diffusion models, the execution process of our modified model approximates the conditional probability distribution under the historical hidden state $\boldsymbol{h}_{t-1}$, which makes our method not only focus on the neighboring states when representing the distribution of the time series, but also take into full consideration of the long-term historical state that is indispensable for modeling the time series. This improvement makes the diffusion model more attuned to the essential needs of spatio-temporal modeling, thus effectively representing the potential temporal distribution. Specifically, we employ an encoder module (e.g., an RNN-based model) to extract temporal information step by step as depicted in Fig. 3. This module results in the acquisition of a hidden state $\boldsymbol{h}$ for each time step, which is rich in historical temporal patterns. We denote the hidden state as $\boldsymbol{h}_{t-1}$ at the (t-1)-th step, which subsequently serves as a conditional input in the diffusion model for the generation of the state at time step t. The entire process can be bifurcated into training and inference phases.

3.2.2 Training Phase

By leveraging the data distribution $q(\boldsymbol{x}_t^0)$ and the hidden state $\boldsymbol{h}_{t-1}$, the objective of the conditional diffusion model is to acquire a distribution $p\theta(\boldsymbol{x}_t^{n-1}|\boldsymbol{x}_t^n, \boldsymbol{h}_{t-1})$ that accurately represents the data distribution $q(\boldsymbol{x}_t^0)$ while maintaining ease of sampling. The forward process is delineated by a Markov chain, where incremental Gaussian noise is systematically introduced to the observation $\boldsymbol{x}_t^0$:

$$q\left(\boldsymbol{x}_t^{1:N} \mid \boldsymbol{x}_t^0\right) = \prod_{n=1}^{N} q\left(\boldsymbol{x}_t^n \mid \boldsymbol{x}_t^{n-1}\right). \tag{3}$$

Here, $q\left(\boldsymbol{x}_t^n \mid \boldsymbol{x}_t^{n-1}\right)$ is modeled as a Gaussian distribution, formalized as:

$$q\left(\boldsymbol{x}_t^n \mid \boldsymbol{x}_t^{n-1}\right) = \mathcal{N}\left(\boldsymbol{x}_t^n; \sqrt{1-\beta_n}\boldsymbol{x}_t^{n-1}, \beta_n \boldsymbol{I}\right), \tag{4}$$

where $\{\beta_1, \cdots, \beta_N\}$ constitutes an ascending variance schedule, with $\beta_n \in (0,1)$ representing the noise level at forward step n. Introducing $\hat{\alpha}_n = 1 - \beta_n$ and $\alpha_n = \prod_{n=1}^{N} \hat{\alpha}_n$, a distinctive feature of the forward process is the existence of a closed-form expression for the distribution of $\boldsymbol{x}_t^n$ given $\boldsymbol{x}_t^0$:

$$q\left(\boldsymbol{x}_t^n \mid \boldsymbol{x}_t^0\right) = \mathcal{N}\left(\boldsymbol{x}_t^n; \sqrt{\alpha_n}\boldsymbol{x}_t^0, (1-\alpha_n)\boldsymbol{I}\right), \tag{5}$$

which is alternatively expressed as $\boldsymbol{x}_t^n = \sqrt{\alpha_n}\boldsymbol{x}_t^0 + \sqrt{1-\alpha_n}\epsilon$ through the reparameteriztioin trick [48], where $\epsilon \sim \mathcal{N}(\boldsymbol{0}; \boldsymbol{I})$ denotes a sampled noise and $\boldsymbol{I}$ is the identity matrix.

The reverse process involves the denoising of $\boldsymbol{x}_t^N$ to reconstruct $\boldsymbol{x}_t^0$ iteratively, guided by the hidden state $\boldsymbol{h}_{t-1}$. This process also adheres to a Markov chain but with learnable Gaussian transitions that commence with $p(\boldsymbol{x}_t^N) = \mathcal{N}(\boldsymbol{x}^N; \mathbf{0}, \boldsymbol{I})$:

$$p_\theta\left(\boldsymbol{x}_t^{0:N} \mid \boldsymbol{h}_{t-1}\right) = p\left(\boldsymbol{x}_t^N\right) \prod_{n=N}^{1} p_\theta\left(\boldsymbol{x}_t^{n-1} \mid \boldsymbol{x}_t^n, \boldsymbol{h}_{t-1}\right). \quad (6)$$

Subsequently, the transition between two nearby latent variables is represented as follows:

$$p_\theta\left(\boldsymbol{x}_t^{n-1} \mid \boldsymbol{x}_t^n, \boldsymbol{h}_{t-1}\right) = \mathcal{N}\left(\boldsymbol{x}_t^{n-1}; \mu_\theta\left(\boldsymbol{x}_t^n, \boldsymbol{h}_{t-1}, n\right), \sigma_\theta\left(\boldsymbol{x}_t^n, n\right)\right), \quad (7)$$

with shared parameters θ. A parameterization procedure [44] is applied to $p_\theta\left(\boldsymbol{x}_t^{n-1} \mid \boldsymbol{x}_t^n, \boldsymbol{h}_{t-1}\right)$:

$$\begin{aligned} \mu_\theta\left(\boldsymbol{x}_t^n, \boldsymbol{h}_{t-1}, n\right) &= \frac{1}{\alpha_n}\left(\boldsymbol{x}_t^n - \frac{\beta_n}{\sqrt{1-\alpha_n}}\epsilon_\theta\left(\boldsymbol{x}_t^n, \boldsymbol{h}_{t-1}, n\right)\right), \\ \sigma_\theta\left(\boldsymbol{x}^n, n\right) &= \frac{1-\alpha_{n-1}}{1-\alpha_n}\beta_n, \end{aligned} \quad (8)$$

where ϵ_θ is a trainable denoising function determining the extent of noise removal at the current denoising step. Noise inference for any stage of the process is computed as:

$$\boldsymbol{x}_t^{n-1} = \frac{1}{\sqrt{\alpha_n}}\left(\boldsymbol{x}_t^n - \frac{\beta_n}{\sqrt{1-\alpha_n}}\epsilon_\theta\left(\boldsymbol{x}_t^n, \boldsymbol{h}_{t-1}, n\right)\right) + \sqrt{\sigma_\theta}\boldsymbol{z}, \quad (9)$$

where $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}; \boldsymbol{I})$ for $n \in \{N, \cdots, 2\}$, and $\boldsymbol{z} = \mathbf{0}$ when n=1. The parameters θ are learned by solving the following optimization problem:

$$arg\min_\theta \mathbb{E}_{\boldsymbol{x}_t^0 \sim q(\boldsymbol{x}_t^0), \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), n} \left\|\epsilon - \epsilon_\theta\left(\boldsymbol{x}_t^n, \boldsymbol{h}_{t-1}, n\right)\right\|_2^2, \quad (10)$$

where the ϵ_θ network is also conditioned on the hidden state. Substituting $\boldsymbol{x}_t^n = \sqrt{\alpha_n}\boldsymbol{x}_t^0 + \sqrt{1-\alpha_n}\epsilon$ into the optimization function yields:

$$\begin{aligned} arg\min_\theta &\mathbb{E}_{\boldsymbol{x}_t^0 \sim q(\boldsymbol{x}_t^0), \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), n} \\ &\left\|\epsilon - \epsilon_\theta\left(\sqrt{\alpha_n}\boldsymbol{x}_t^0 + \sqrt{1-\alpha_n}\epsilon, \boldsymbol{h}_{t-1}, n\right)\right\|_2^2. \end{aligned} \quad (11)$$

The training procedure for each time step is outlined in Algorithm 1. As shown in Eq. (11), the training process of the proposed conditional diffusion model with an optimization objective is to minimize the difference between the sampled noise and the learnable noise function, so that the final learned distribution information is closely approximated to the real distribution. Therefore, the convergence condition of the optimization objective expressed in Eq. (11) is that $\mathbb{E}_{\boldsymbol{x}_t^0 \sim q(\boldsymbol{x}_t^0), \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), n}$ should stabilize and reach a minimum value.

3.2.3 Inference Phase

Upon completion of training, the model, having acquired knowledge of potential historical temporal distribution information, possesses the capacity to generate temporal patterns at any timestep by sampling from a Gaussian distribution conditioned on the historical hidden state. Consequently, the model can be leveraged to infer the corresponding temporal prompts, which imply the evolving temporal trends in spatio-temporal data. As delineated in Algorithm 2, the inference process employs the trained denoising function ϵ_θ to iteratively sample the temporal prompts $\boldsymbol{p}_t$ based on $\boldsymbol{h}_{t-1}$ in accordance with Eq. 6.

Algorithm 1 Training of conditional diffusion model.

Input: data $\boldsymbol{x}_t^0 \sim q(\boldsymbol{x}_t^0)$ and hidden state $\boldsymbol{h}_{t-1}$.
Output: Trained denoising function ϵ_θ.
repeat
1: Initialize $n \sim$ Uniform$(1, \cdots, N)$ and $\epsilon \in \mathcal{N}(\mathbf{0}; \boldsymbol{I})$.
2: Take gradient step on $\nabla_\theta \left\|\epsilon - \epsilon_\theta\left(\sqrt{\alpha_n}\boldsymbol{x}_t^0 + \sqrt{1-\alpha_n}\epsilon, \boldsymbol{h}_{t-1}, n\right)\right\|_2^2$.
until converged.

Algorithm 2 Inference of conditional diffusion model.

Input: temporal prompt $\boldsymbol{p}_t^N \sim \mathcal{N}(\mathbf{0}; \boldsymbol{I})$ and hidden state $\boldsymbol{h}_{t-1}$.
1: **for** $n = N$ to 1 **do**
2: **if** n>1 **then** $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}; \boldsymbol{I})$
3: **else** $\boldsymbol{z} = 0$
4: **end if** $\boldsymbol{p}_t^{n-1} = \frac{1}{\sqrt{\alpha_n}}\left(\boldsymbol{p}_t^n - \frac{\beta_n}{\sqrt{1-\alpha_n}}\epsilon_\theta\left(\boldsymbol{p}_t^n, \boldsymbol{h}_{t-1}, n\right)\right) + \sqrt{\sigma_\theta}\boldsymbol{z}$
5: **end for**
6: **return** $\boldsymbol{p}_t^0$

3.3 Spatial Prompt Representation

As Fig. 2 (a) illustrates, topological connectivity such as geographic proximity establishes intrinsic inter-node relationships in space, preventing nodes from existing in isolation. Several neighboring nodes with high connectivity may collectively form a subgraph with a distinctive aggregation pattern. Consequently, these multi-grained subgraph patterns serve as a priori guidance in revealing the internal node pattern, and the instructive information from subgraphs to nodes inherent in spatial distribution, is denoted as "spatial prompts". However, prevailing spatial modeling methods predominantly capture spatial correlations at a single granularity, lacking the capacity to effectively model intricate hierarchical structures [49], [50].

Motivated by recent advancements in GNNs, we recognize the pivotal role of graph pooling in hierarchical graph representation learning [51], [52]. This technique facilitates a reasonable partition of subgraph structures end-to-end based on the adjacency information and node features within the graph signal [53], [54]. Drawing on this concept, we construct a structure-aware graph derived from the original graph, capable of adaptively representing diversified subgraph structures. This approach lays the foundation for subsequent extraction of structure distribution to generate spatial prompts.

3.3.1 Adaptive Spatial Structure Partition

In the context delineated above, our initial step involves the design of an adaptive assignment matrix for spatial structure partitioning, whose understanding of spatial structure stems from learning the original graph signal with the

adjacency matrix $\boldsymbol{A}$ and feature embedding $\boldsymbol{h}_c$ through a GNN module, expressed as:

$$\begin{aligned} \bar{\boldsymbol{h}_c} &= \mathrm{GNN}\left(\boldsymbol{h}_c, \tilde{\boldsymbol{A}}\right), \\ \boldsymbol{S} &= \mathrm{Softmax}(\mathrm{ReLU}(\bar{\boldsymbol{h}_c}\boldsymbol{W}_{s,1})\boldsymbol{W}_{s,2}), \end{aligned} \tag{12}$$

where $\boldsymbol{S} \in \mathbb{R}^{N \times N^s}$ is the learnable subgraph assignment matrix, N^s is the number of subgraphs, $\tilde{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} \in \mathbb{R}^{N \times N}$ is the symmetrically normalized adjacency matrix, and $\boldsymbol{D} = \mathrm{diag}\left(\boldsymbol{A}\boldsymbol{1}_N\right)$ is the degree matrix.

Notably, the assignment matrix $\boldsymbol{S}$ is essentially a probabilistic assignment matrix, and $\boldsymbol{S}(i,j)$ represents the probability that node i is aggregated to subgraph j. Based on the assignment matrix, the adjacency matrix $\boldsymbol{A}^s$ of the structure-aware graph G^s is obtained as:

$$\boldsymbol{A}^s = \boldsymbol{S}^\top \boldsymbol{A}\boldsymbol{S}. \tag{13}$$

Besides, we can pool the temporal embeddings $\boldsymbol{h}_c$ from the clients' encoders to obtain the embedding matrix $\boldsymbol{h}_c^s$ of G^s based on the learned assignment matrix:

$$\boldsymbol{h}_c^s = \boldsymbol{S}^\top \boldsymbol{h}_c. \tag{14}$$

Thereby, a structure-aware graph containing the structure distribution patterns of the spatio-temporal graph can be adaptively constructed.

To encourage that nodes with larger connection strength are assigned to the same subgraph, we utilize the learned assignment matrix $\boldsymbol{S}$ to reconstruct the adjacency matrix $\boldsymbol{A}$, and then optimize the connectivity loss as follows,

$$\begin{aligned} \mathcal{L}_{con} = &\frac{1}{N^2} \sum_{i,j \in V} -\boldsymbol{A}[i,j] \log\left(\hat{\boldsymbol{A}}[i,j]\right) \\ &- (1 - \boldsymbol{A}[i,j]) \log\left(1 - \hat{\boldsymbol{A}}[i,j]\right), \\ \text{where} \quad &\hat{\boldsymbol{A}} = \mathrm{Sigmoid}(\boldsymbol{S}\boldsymbol{S}^\top). \end{aligned} \tag{15}$$

The elements in $\boldsymbol{A}$ represent the connection strength between two nodes, and elements in $\hat{\boldsymbol{A}}$ represent the probabilities that two nodes are divided into the same subgraph.

Furthermore, to discover representative and diverse spatio-temporal patterns in G^s, we integrate an orthogonal loss to diversify subgraph representations and minimize the cosine similarity of each node pair, which is formalized as,

$$\mathcal{L}_{ort} = \frac{1}{C_{N^s}^2} \sum_{i=1}^{N^s} \sum_{j=i+1}^{N^s} \left| \frac{\boldsymbol{h}_c^s[i] \odot \boldsymbol{h}_c^s[j]}{\|\boldsymbol{h}_c^s[i]\| \cdot \|\boldsymbol{h}_c^s[j]\|} \right|, \tag{16}$$

where $C_{N^s}^2$ is the number of 2-combinations of N^s. Both regularization terms are integrated into the final objective function to achieve more reasonable subgraph assignments.

3.3.2 Spatial Prompt Generation

Based on the aforementioned two spatial granularities of spatio-temporal graphs, we employ Graph Convolutional Networks (GCNs) [55] to capture multi-grained spatial dependencies in parallel, which is widely adopted in spatio-temporal mobility modeling:

$$\begin{aligned} \boldsymbol{h}_g &= \mathrm{Sigmoid}((\boldsymbol{A} + \boldsymbol{I})\boldsymbol{h}_c^{(l)}\boldsymbol{W}_1^{(l)} + \boldsymbol{b}_1^{(l)}), \\ \boldsymbol{h}_g^s &= \mathrm{Sigmoid}((\boldsymbol{A}^s + \boldsymbol{I}^s)\boldsymbol{h}_c^{s(l)}\boldsymbol{W}_2^{(l)} + \boldsymbol{b}_2^{(l)}), \end{aligned} \tag{17}$$

where $\boldsymbol{h}_g$ and $\boldsymbol{h}_g^s$ are the spatial embeddings after the message propagation of GCNs, and $\boldsymbol{I}, \boldsymbol{I}^s$ are the identity matrices. Notice that, the spatial information in $\boldsymbol{h}_g$ is the client-specific pattern for each node, while the information in $\boldsymbol{h}_g^s$ contains the macroscopic structured patterns for the subgraphs.

Subsequently, we extract beneficial spatial prompts from the aforementioned macroscopic structured patterns for each client. We employ the attention mechanism [56] to quantify the intimate interaction between nodes and subgraphs, which requires computing the query vector $\boldsymbol{Q}$, key vector $\boldsymbol{K}$, and value vector $\boldsymbol{V}$ based on $\boldsymbol{h}_g$ and $\boldsymbol{h}_g^s$:

$$\boldsymbol{Q} = \boldsymbol{h}_g\boldsymbol{W}_k, \quad \boldsymbol{K} = \boldsymbol{h}_g^s\boldsymbol{W}_k, \quad \boldsymbol{V} = \boldsymbol{h}_g^s\boldsymbol{W}_v. \tag{18}$$

Then, the attention matrix is obtained by calculating $\boldsymbol{Q}\boldsymbol{K}^\top$, quantifying the degree of influence between subgraphs and nodes. Intuitively, the spatial prompts $\boldsymbol{p}_s$ are derived after the interaction through the standard attention mechanism, which is expressed as:

$$\boldsymbol{p}_s = \mathrm{Softmax}\left(\boldsymbol{Q}\boldsymbol{K}^\top / \sqrt{D}\right)\boldsymbol{V}. \tag{19}$$

However, softmax attention computes the similarity between all query-key pairs, leading to quadratic complexity and constraining the scalability of the model [57]. To address this issue, we introduce a linear attention mechanism [58] as a replacement for softmax attention. The linear attention applies a kernel-based mapping function $\phi(\cdot)$ to $\boldsymbol{Q}$ and $\boldsymbol{K}$ respectively, altering the computation order and substantially reducing complexity to a linear level. Consequently, the process of spatial prompt generation in Eq. 19 can be transformed into an efficient form:

$$\boldsymbol{p}_s = \left(\phi(\boldsymbol{Q})\phi(\boldsymbol{K})^\top\right)\boldsymbol{V} = \phi(\boldsymbol{Q})\left(\phi(\boldsymbol{K})^\top\boldsymbol{V}\right). \tag{20}$$

Note that, we utilize the exponential linear unit activation function $\mathrm{ELU}(\cdot)$ to implement the feature map $\phi(\cdot)$, which is applied row-wise to the matrices $\boldsymbol{Q}$ and $\boldsymbol{K}$. Specifically, carefully designed mapping functions $\phi(\cdot)$ are applied to $\boldsymbol{Q}$ and $\boldsymbol{K}$ respectively, i.e., $\phi(\boldsymbol{Q})\phi(\boldsymbol{K})^\top$. This allows us to change the computation order from $(\phi(\boldsymbol{Q})\phi(\boldsymbol{K})^\top)\boldsymbol{V}$ to $\phi(\boldsymbol{Q})\left(\phi(\boldsymbol{K})^\top\boldsymbol{V}\right)$ based on the associative property of matrix multiplication, so that we can compute $\sum_{j=1}^{N}\phi\left(\boldsymbol{K}_j\right)\boldsymbol{V}_j^\top$ and $\sum_{j=1}^{N}\phi\left(\boldsymbol{K}_j\right)$ once and reuse them for every query. By doing so, the computation complexity for the token number is reduced to $\mathcal{O}(N)$ [58].

The spatial prompts play a pivotal role in accurately guiding the models to discover the distinct latent spatial distribution characteristics in clients, thereby endowing the models with personalized structure information.

3.4 Multi-view Feature Fusion

As illustrated in Fig. 3, we concatenate the outputs of the client-side encoder $\boldsymbol{h}_{c,i}$ and the server-side GCNs $\boldsymbol{h}_{g,i}$, along with two distinct spatio-temporal prompts $\boldsymbol{p}_{t,i}$, $\boldsymbol{p}_{s,i}$. Subsequently, joint features are derived, containing latent spatio-temporal distribution information suitable for local predictions.

$$\boldsymbol{o}_i = \mathrm{Concatenate}\left(\boldsymbol{h}_{c,i}, \boldsymbol{h}_{g,i}, \boldsymbol{p}_{t,i}, \boldsymbol{p}_{s,i}\right). \tag{21}$$

Intuitively, the simple concatenation of the aforementioned features overlooks the spatio-temporal interactions

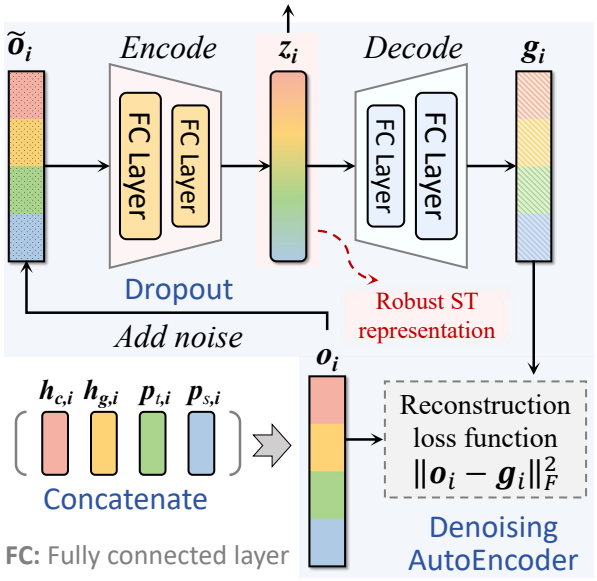

Fig. 5. Detailed process of multi-view feature fusion with a Denoising AutoEncoder. Multi-view features are fused by optimizing a reconstruction loss function after adding noise and reconstruction to obtain robust and personalized ST representations for local tasks.

among them. To acquire a robust spatio-temporal representation, we further incorporate a Denoising AutoEncoder [59] (DAE). The DAE is employed to capture the intricate relationships and generate robust feature representations through the use of dropout layers instead of random noise. In detail, we first input the concatenated features $\boldsymbol{o}_i$ into a dropout layer to introduce noise, which is formalized as:

$$\tilde{\boldsymbol{o}_i} = \text{Dropout}(\boldsymbol{o}_i). \tag{22}$$

Subsequently, we encode $\tilde{\boldsymbol{o}_i}$ using two fully connected layers to obtain the compressed hidden feature:

$$\begin{aligned} \tilde{\boldsymbol{o}}_{1,i} &= \text{ReLU}\left(\boldsymbol{W}_1\tilde{\boldsymbol{o}_i} + \boldsymbol{b}_1\right), \\ \boldsymbol{z}_i &= \text{ReLU}\left(\boldsymbol{W}_2\tilde{\boldsymbol{o}}_1 + \boldsymbol{b}_2\right). \end{aligned} \tag{23}$$

Next, we similarly reconstruct $\boldsymbol{z}_i$ by decoding it through two fully connected layers:

$$\begin{aligned} \tilde{\boldsymbol{z}}_{1,i} &= \text{ReLU}\left(\boldsymbol{W}_3\tilde{\boldsymbol{z}_i} + \boldsymbol{b}_3\right), \\ \boldsymbol{g}_i &= \text{ReLU}\left(\boldsymbol{W}_4\tilde{\boldsymbol{z}}_{1,i} + \boldsymbol{b}_4\right), \end{aligned} \tag{24}$$

where $\boldsymbol{W}_1 \sim \boldsymbol{W}_4$ and $\boldsymbol{b}_1 \sim \boldsymbol{b}_4$ are the learnable parameters. The reconstruction loss between $\boldsymbol{o}_i$ and the reconstructed feature $\boldsymbol{g}_i$ is calculated using the Frobenius norm, which is formalized as:

$$\mathcal{L}_{rec} = \|\boldsymbol{o}_i - \boldsymbol{g}_i\|_F^2, \tag{25}$$

where $\|\cdot\|_F^2$ represents the Frobenius norm. Here, we select the hidden feature $\boldsymbol{z}_i$ encapsulating spatio-temporal interaction information as the joint distribution feature. The detailed process of multi-view feature fusion is depicted in Fig. 5.

Finally, we feed $\boldsymbol{z}_i$ into the client-side decoder, and a client-side predictor, composed of a fully connected layer, is incorporated to ensure that the output maintains the same dimension and shape as the forecasting target $\hat{\boldsymbol{x}}_i$. The final fully connected layer employs the ReLU activation function. We utilize Mean Absolute Errors to measure the disparity between the predicted values and ground-truth to optimize the performance of the local model.

3.5 Federated Aggregation

Following the local model training, the client is required to transmit the local parameters to the server for global aggregation, enabling parameter updates within the distributed architecture. Notably, we restrict the update to partial parameters of the client-side model, encompassing the encoder, decoder, diffusion module, and DAE. The predictor is employed to disentangle a client-specific distribution of local data, where parameters differ across clients. Consequently, it is imperative to preserve the personalization of parameters in the predictor to better conform to the local data distribution. Specifically, we employ the FedAvg algorithm [60], a prominent algorithm in the federated learning domain, to achieve global aggregation. The detailed process of global aggregation is formalized as:

$$\omega_{t+1} \leftarrow \sum_{i=1}^{N} \frac{|\mathcal{D}^i|}{|\mathcal{D}|}\omega_{i,t}, \tag{26}$$

where t is the number of training rounds. Additionally, our FedCroST can be seamlessly integrated with other aggregation algorithms, showcasing considerable scalability, as discussed in Section 4.4.3.

3.6 Theoretical Analysis

In this section, we provide theoretical analyses to substantiate the feasibility of our model.

3.6.1 Optimization in Training Diffusion

To achieve effective generation, the training process of the diffusion model is designed to minimize the gap between distributions $p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n)$ and $q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n, \boldsymbol{x}^0)$, i.e., minimize the negative log-likelihood $-logp_\theta(\boldsymbol{x}^0)$. We omit t for succinctness. The proof of this assertion is as follows,

Proof 1. First, we compute the upper bound of $-logp_\theta(\boldsymbol{x}^0)$:

$$\begin{aligned} &-logp_\theta(\boldsymbol{x}^0) \le -logp_\theta(\boldsymbol{x}^0) + D_{KL}(q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)||p_\theta(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)) \\ &= -logp_\theta(\boldsymbol{x}^0) + \mathbb{E}_{\boldsymbol{x}^{1:N}\sim q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}[log\frac{q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{0:N})/p_\theta(\boldsymbol{x}^0)}] \\ &= -logp_\theta(\boldsymbol{x}^0) + \mathbb{E}_{\boldsymbol{x}^{1:N}\sim q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}[log\frac{q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{0:N})} + logp_\theta(\boldsymbol{x}^0)]. \end{aligned} \tag{27}$$

Taking the expectation with respect to $\boldsymbol{x}^0$ on both sides yields:

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}^0\sim q(\boldsymbol{x}^0)}[-logp_\theta(\boldsymbol{x}^0)] \le \mathbb{E}_{\boldsymbol{x}^0\sim q(\boldsymbol{x}^0)}[-logp_\theta(\boldsymbol{x}^0) \\ &+ \mathbb{E}_{\boldsymbol{x}^{1:N}\sim q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}[log\frac{q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{0:N})} + logp_\theta(\boldsymbol{x}^0)]] \\ &= \mathbb{E}_{\boldsymbol{x}^0\sim q(\boldsymbol{x}^0)}[-logp_\theta(\boldsymbol{x}^0)] \\ &+ \mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}_{0:N})}[log\frac{q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{0:N})} + logp_\theta(\boldsymbol{x}^0)] \\ &= \mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}^{0:N})}[log\frac{q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{0:N})}]. \end{aligned} \tag{28}$$

To minimize the above mathematical expectation, we adopt the Markov hypothesis to obtain:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}^{0:N})}[log\frac{q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{0:N})}]\\
&=\mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}^{0:N})}[log\frac{\prod_{n=1}^{N}q(\boldsymbol{x}^n|\boldsymbol{x}^{n-1})}{p_\theta(\boldsymbol{x}^N)\prod_{n=1}^{N}p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n)}]\\
&=\mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}^{0:N})}[-logp_\theta(\boldsymbol{x}^N)+log\frac{\prod_{n=1}^{N}q(\boldsymbol{x}^n|\boldsymbol{x}^{n-1})}{\prod_{n=1}^{N}p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n)}]\\
&=\mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}^{0:N})}[-logp_\theta(\boldsymbol{x}^N)\\
&+\sum_{n=2}^{N}log\frac{q(\boldsymbol{x}^n|\boldsymbol{x}^{n-1})}{p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n)}+log\frac{q(\boldsymbol{x}^1|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^0|\boldsymbol{x}^1)}].
\end{aligned}\tag{29}$$

Next, we decompose $q(\boldsymbol{x}^n|\boldsymbol{x}^{n-1})$ as follows:

$$\begin{aligned}
&q(\boldsymbol{x}^n|\boldsymbol{x}^{n-1})=q(\boldsymbol{x}^n|\boldsymbol{x}^{n-1},\boldsymbol{x}^0)=\frac{q(\boldsymbol{x}^n,\boldsymbol{x}^{n-1},\boldsymbol{x}^0)}{q(\boldsymbol{x}^{n-1},\boldsymbol{x}^0)}\\
&=\frac{q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n,\boldsymbol{x}^0)q(\boldsymbol{x}^n|\boldsymbol{x}^0)q(\boldsymbol{x}^0)}{q(\boldsymbol{x}^{n-1},\boldsymbol{x}^0)}=\frac{q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n,\boldsymbol{x}^0)q(\boldsymbol{x}^n|\boldsymbol{x}^0)}{q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^0)}.
\end{aligned}\tag{30}$$

Then, we substitute Eq. 30 into Eq. 29:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}^{0:N})}[log\frac{q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{0:N})}]=\mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}^{0:N})}[-logp_\theta(\boldsymbol{x}^n)\\
&+\sum_{n=2}^{N}log\frac{q(\boldsymbol{x}^n|\boldsymbol{x}^{n-1})}{p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n)}+log\frac{q(\boldsymbol{x}^1|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^0|\boldsymbol{x}^1)}]\\
&=\mathbb{E}_{\boldsymbol{x}^{0:N}\sim q(\boldsymbol{x}^{0:N})}[-logp_\theta(\boldsymbol{x}^n)+\sum_{n=2}^{N}log\frac{q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n,\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n)}\\
&=\mathbb{E}_{\boldsymbol{x}^0\sim q(\boldsymbol{x}^0)}[\mathbb{E}_{\boldsymbol{x}^{1:N}\sim q(\boldsymbol{x}^{1:N}|\boldsymbol{x}^0)}[log\frac{q(\boldsymbol{x}^n|\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^n)}\\
&+\sum_{n=2}^{N}log\frac{q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n,\boldsymbol{x}^0)}{p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n)}-logp_\theta(\boldsymbol{x}^0|\boldsymbol{x}^1)]]\\
&=\mathbb{E}_{\boldsymbol{x}^0\sim q(\boldsymbol{x}^0)}[D_{KL}(q(\boldsymbol{x}^n|\boldsymbol{x}^0)||p_\theta(\boldsymbol{x}^n))\\
&+\sum_{n=2}^{N}D_{KL}(q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n,\boldsymbol{x}^0)||p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n))-logp_\theta(\boldsymbol{x}^0|\boldsymbol{x}^1)].
\end{aligned}\tag{31}$$

For $\sum_{n=2}^{N}D_{KL}(q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n,\boldsymbol{x}^0)||p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n))$, each term calculates the KL-divergence between $q(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n,\boldsymbol{x}^0)$ and $p_\theta(\boldsymbol{x}^{n-1}|\boldsymbol{x}^n))$. Hence, the more similar the two distributions, the smaller $KL(\cdot,\cdot)$ and the negative log-likelihood.

3.6.2 Convergence Analysis

We provide insights into the convergence analysis of FedCroST. We denote the local objective function as $\mathcal{L}$ with a subscript signifying the iteration count and make the following assumptions inherited from previous work [61], [62].

Assumption 1. (Lipschitz smooth). Each local objective function is L_1-Lipschitz smooth, signifying that the gradient of local objective function is L_1-Lipschitz continuous,

$$\begin{aligned}
&\|\nabla\mathcal{L}_{t_1}-\nabla\mathcal{L}_{t_2}\|_2\le L_1\|\boldsymbol{\omega}_{i,t_1}-\boldsymbol{\omega}_{i,t_2}\|_2,\\
&\forall t_1,t_2>0,i\in[N].
\end{aligned}\tag{32}$$

This assumption implies the following quadratic bound,

$$\begin{aligned}
&\mathcal{L}_{t_1}-\mathcal{L}_{t_2}\le\langle\nabla\mathcal{L}_{t_2},(\boldsymbol{\omega}_{i,t_1}-\boldsymbol{\omega}_{i,t_2})\rangle+\frac{L_1}{2}\|\boldsymbol{\omega}_{i,t_1}-\boldsymbol{\omega}_{i,t_2}\|_2^2,\\
&\forall t_1,t_2>0,i\in[N].
\end{aligned}\tag{33}$$

Assumption 2. (Bounded data heterogeneity). The stochastic gradient $g_{i,t}=\nabla\mathcal{L}(\boldsymbol{\omega}_t,\xi_t)$ is an unbiased estimator of the local gradient for each client. Suppose its expectation

$$\mathbb{E}_{\xi_i\sim D_i}[g_{i,t}]=\nabla\mathcal{L}(\boldsymbol{\omega}_{i,t})=\nabla\mathcal{L}_t,\forall i\in[N],\tag{34}$$

and its variance is bounded by constant σ^2:

$$\mathbb{E}[\|g_{i,t}-\nabla\mathcal{L}(\boldsymbol{\omega}_{i,t})\|_2^2]\le\sigma^2,\forall i\in[N],\sigma\ge0,\tag{35}$$

which reflects the degree of heterogeneity in the data distribution across clients.

Assumption 3. (Bounded expectation of euclidean norm of stochastic gradients). The expectation of the stochastic gradient is bounded by G:

$$\mathbb{E}[\|g_{i,t}\|_2^2]\le G^2,\forall i\in[N],G\ge0.\tag{36}$$

Building upon these assumptions, we present theoretical results for the non-convex problem. To assess the convergence of the proposed FedCroST, we employ the technique introduced in [63] to define:

$$\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\omega}_{t^*})\|_2^2]:=\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\omega}_t)\|_2^2],\tag{37}$$

where t^* is uniformly sampled from the set $\{0,\cdots,T-1\}$.

Note that even though the aggregated model $\boldsymbol{\omega}_t$ are not computed at each time t, there is virtual averaged model $\boldsymbol{\omega}_t$ whose corresponding averaged update is:

$$\boldsymbol{\omega}_{t+1}\leftarrow\boldsymbol{\omega}_t-\eta\boldsymbol{g}_t,\quad\boldsymbol{g}_t:=\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{g}_{k,t}.\tag{38}$$

Based on Assumption 1 about Lipschitz smoothness:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}\left(\boldsymbol{\omega}_{t+1}\right)\right]\le\mathbb{E}\Big[&\mathcal{L}\left(\boldsymbol{\omega}_t\right)+\nabla\mathcal{L}\left(\boldsymbol{\omega}_t\right)^\top\left(\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}_t\right)\\
&+\frac{L_1}{2}\left\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}_t\right\|^2\Big].
\end{aligned}\tag{39}$$

Theorem 1. (Convergence of FedCroST). Considering the optimization objective under Assumption 1, 2, 3, if the total number of communication rounds T is pre-defined, the following convergence result holds:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla\mathcal{L}\left(\boldsymbol{\omega}_t\right)\right\|_2^2\right]\le\frac{2}{\eta}\mathbb{E}\left[\mathcal{L}\left(\boldsymbol{\omega}_{t+1}\right)-\mathcal{L}\left(\boldsymbol{\omega}_t\right)\right]+\left(\eta L_1-1\right)\mathbb{E}\left[\left\|\boldsymbol{g}_t\right\|^2\right]\\
&+\frac{2}{N}{L_1}^2\sum_{k=1}^{N}\mathbb{E}\left[\left\|\boldsymbol{\omega}_t-\boldsymbol{\omega}_{k,t}\right\|^2\right]+\frac{2}{N}\sum_{k=1}^{N}\mathbb{E}\left[\left\|\nabla\mathcal{L}_k\left(\boldsymbol{\omega}_{k,t}\right)-\boldsymbol{g}_{k,t}\right\|^2\right].
\end{aligned}\tag{40}$$

Average the results of each round:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\omega}_t)\|_2^2]\le\frac{2}{\eta T}\mathbb{E}\left[\mathcal{L}\left(\boldsymbol{\omega}_0\right)-\mathcal{L}\left(\boldsymbol{\omega}_t\right)\right]\\
&+\frac{(\eta L_1-1)}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\boldsymbol{g}_t\right\|^2\right]+\frac{2}{NT}{L_1}^2\sum_{t=0}^{T-1}\sum_{k=1}^{N}\mathbb{E}\left[\left\|\boldsymbol{\omega}_t-\boldsymbol{\omega}_{k,t}\right\|^2\right]\\
&+\frac{2}{NT}\sum_{t=0}^{T-1}\sum_{k=1}^{N}\mathbb{E}\left[\left\|\nabla\mathcal{L}_k\left(\boldsymbol{\omega}_{k,t}\right)-\boldsymbol{g}_{k,t}\right\|^2\right]\\
&=\mathcal{O}\left(\frac{1}{\eta T}\right)+\mathcal{O}\left(\frac{E^2\eta^2G^2}{N}\right)+\mathcal{O}\left(E^2\eta^2\sigma^2\right)+\mathcal{O}\left(\frac{\lambda\mathcal{L}_2E^2\eta^2}{N}\right).
\end{aligned}\tag{41}$$

Theorem 1 indicates that convergence can be guaranteed by choosing the appropriate learning rate η and the importance weight λ.

Corollary 1. (Convergence setting). Under the setting of Theorem 1, suppose the learning rate is chosen as $\eta = \frac{\sqrt{N}}{\sqrt{T}}$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\omega}_t)\|_2^2] \leq \mathcal{O}\left(\frac{1}{\sqrt{NT}}\right) + \mathcal{O}\left(\frac{E^2G^2}{T}\right) + \mathcal{O}\left(\frac{NE^2\sigma^2}{T}\right) + \mathcal{O}\left(\frac{\lambda\mathcal{L}_2E^2}{T}\right). \tag{42}$$

Thus, the loss function converges after carefully selecting the number of communication rounds T and hyperparameter λ.

Remark 1. (Convergence analysis). Corollary 1 provides a convergence setting for Theorem 1. The smaller the bound, the larger T is, which reflects the numerical correlation between the convergence boundary and the number of communication rounds, and provides a theoretical basis for the setting of learning parameters.

3.7 Summary

Our proposed method is dedicated to modeling ST mobility in the often-overlooked yet increasingly prevalent data silo scenario. To highlight our contributions more clearly, we further summarize the innovative points and theoretical arguments of our work as follows: **i) Innovative points:** To investigate the significance of latent ST data distributions in enabling personalized federated ST learning, we propose two effective distributional representation modules from temporal and spatial dimensions, respectively. From the temporal dimension, we devise a hidden state-based conditional diffusion module to reveal potential evolutionary characteristics within the time series. From the spatial dimension, we introduce an adaptive spatial structure partitioning method to capture inherent structural distributions within node neighborhoods. Additionally, we ingeniously integrate multi-view ST information through a denoising autoencoder to obtain robust and personalized ST representations for local tasks. Therefore, these innovative technical designs provide a viable implementation path for cross-silo ST modeling. **ii) Theoretical arguments:** To further validate the feasibility of our approach, we carry out theoretical derivations from two perspectives: model optimization and model convergence. Firstly, for the training of the diffusion model, we demonstrate that the learned characteristics of the time-series distribution can closely approximate the real distribution. Secondly, for the convergence of the federated system, we derive the upper bound of convergence and the convergence conditions of the model from the local optimization objective. Through multifaceted analysis, we provide strong theoretical arguments for the proposed method.

4 Experiments

In this section, we conduct comprehensive experiments on two real-world datasets in spatio-temporal mobility forecasting from multiple perspectives, including performance comparisons, ablation studies, hyperparameter studies, generalization studies, scalability studies, and visual analysis.

4.1 Experimental Settings

4.1.1 Tasks & Datasets

To investigate the effectiveness of the proposed FedCroST, we evaluate our methods and baselines on two real-world datasets in spatio-temporal mobility forecasting task: 1) **Air quality forecasting**: KnowAir [1] is a $PM_{2.5}$ concentration dataset that covers a total of 184 cities (nodes) in China [64]. 2) **Epidemic spread forecasting**: Covid-TX [2] is a COVID-19 spread dataset that covers 251 counties (hospitalizations) in Texas [65]. Detailed information is given in Table 2. We adopt Z-score normalization to process the data, which standardizes the features by removing the mean and scaling to unit variance.

TABLE 2
Detailed Information of Datasets.

Properties	Datasets	
	KnowAir	Covid-TX
Data Type	Air quality	Epidemic spread
# of Nodes	184	251
# of Time span	01/2017-12/2017	01/2020-12/2020
Time step	3 hour	1 day

4.1.2 Implementation Details

In the experiment, we use P=12 historical time steps to predict the target series over the next Q=12 time steps on both KnowAir and Covid-TX datasets. Each node in the datasets is regarded as an isolated data silo, i.e., a node in the spatio-temporal graph. We utilize the exponential linear unit activation function $\mathrm{ELU}(\cdot)$ to implement the feature map $\phi(\cdot)$, which is defined as $\phi(x) = \mathrm{ELU}(x) + 1$, where $\mathrm{ELU}(\cdot)$ denotes the exponential linear unit activation function [66]. We adopt the alternating optimization in split learning following [30], [67] to train our model in the federated learning phase. The encoder and decoder in the client-side model consist of two layers of an LSTM, and the hidden layer size is 64. We train the model via SGD using Adam optimizer with a learning rate of 0.001. The number of diffusion steps is 100, and the variance schedule is from β_1=1e-4 till β_N=0.1. The denoising function ϵ_θ in the diffusion module is implemented by GRU. The GNN model in Eq. 12 is built on top of the GraphSAGE [68] architecture. All experiments are conducted on NVIDIA A100 GPUs with 40G. We choose the optimal settings through hyperparameter experiments (e.g., N^s) in Section 4.4.5. In addition, in order to verify the impact of the components in our framework, we also design three variants, whose settings are the same as those of our model, except for the corresponding component.

4.1.3 Evaluation Metrics

Two kinds of evaluation metrics are adopted to evaluate the performance of each method, including Root Mean Squared

1. https://github.com/shuowang-ai/PM2.5-GNN
2. https://www.dropbox.com/sh/n0ajd5l0tdeyb80/AABGn-ejfV1YtRwjf_L0AOsNa?dl=0

Errors (**RMSE**) and Mean Absolute Errors (**MAE**), whose detailed definitions are as follows.

$$\begin{aligned} \text{MAE} &= \frac{1}{N \times Q} \sum_{i=1}^{N} \sum_{j=1}^{Q} |\hat{\boldsymbol{x}}_{i,P+j} - \boldsymbol{x}_{i,P+j}| \\ \text{RMSE} &= \sqrt{\frac{1}{N \times Q} \sum_{i=1}^{N} \sum_{j=1}^{Q} (\hat{\boldsymbol{x}}_{i,P+j} - \boldsymbol{x}_{i,P+j})^2}. \end{aligned} \tag{43}$$

Smaller values indicate better performance for the two terms of metrics above.

4.1.4 Baselines

We compare our FedCroST with the related methods, which are categorized into three paradigms: separated, centralized, and federated. The methods involved in the experiments are as follows.

a. Separated training methods: The separated training methods independently model the local data on each client, and the training processes are not related to each other. Therefore, the approaches do not involve the spatial attributes of the spatio-temporal data and generally employ traditional time series modeling methods.

- Autoregressive Integrated Moving Average (**ARIMA**) [69] combines moving averages with autoregression to perform prediction in time series.
- Gated Recurrent Unit (**GRU**) [70] is a deep learning model based on a gating mechanism, which is often used to predict time series data.

b. Centralized training methods: The centralized training methods centrally collect and train data of all nodes, and relevant advanced methods are generally based on GNNs.

- Spatio-temporal Graph Convolutional Networks (**STGCN**) [20] applies ChebNet graph convolution and 1D convolution to extract spatial dependencies and temporal correlations in spatiotemporal data.
- Graph WaveNet (**GWNet**) [24] captures long-range temporal sequences with a stacked dilated 1D convolution and learns a self-adaptive adjacency matrix to capture the spatial dependency.

c. Federated training methods: The federated training methods are the representative baselines we primarily compare, which fit the data silo scenario studied in this paper.

- Federated Learning-based Gated Recurrent Unit neural network (**FedGRU**) [25] integrates FL with a GRU for spatio-temporal forecasting tasks through a local training model without raw data exchange.
- Federated attention-based spatial-temporal graph neural networks (**FASTGNN**) [26] integrates an FL strategy towards topological information protection and a GNN-based model for traffic speed forecasting.
- Federated Deep Learning based on the Spatial-Temporal Long and Short-Term Networks (**FedSTN**) [29] proposes a federated deep learning based on the spatial-temporal long and short-term networks to predict traffic flow.
- Structured Federated Learning framework (**SFL**) [32] is a personalized federated learning framework that leverages the graph structure and discovers comprehensive hidden relationships amongst clients.
- Cross-Node Federated Graph Neural Network (**CNFGNN**) [30] encodes the underlying graph structure using GNN-based architecture under the constraint of cross-node federated learning, which requires that data is generated locally on each node.

TABLE 3
Average performance comparison (MAE and RMSE) of different methods on **KnowAir** (*Air quality forecasting*) and **Covid-TX** (*Epidemic spread forecasting*).

Datasets	Methods	Settings	6-step ahead		12-step ahead	
			MAE	RMSE	MAE	RMSE
KnowAir	ARIMA	Separated	28.45	36.91	36.42	47.71
	GRU	Separated	25.46	32.82	33.57	42.33
	STGCN	Centralized	20.69	27.49	27.72	37.34
	GWNet	Centralized	20.41	26.59	27.61	35.51
	FedGRU	Federated	22.98	31.71	31.41	41.50
	FASTGNN	Federated	22.87	30.86	31.33	41.09
	FedSTN	Federated	22.64	30.62	31.21	40.91
	SFL	Federated	22.43	30.63	31.03	40.88
	CNFGNN	Federated	22.17	29.73	30.71	39.76
	FedCroST (ours)	Federated	**20.74**	**28.29**	**29.28**	**38.15**
Covid-TX	ARIMA	Separated	49.85	79.32	69.22	108.49
	GRU	Separated	46.23	74.56	65.12	99.30
	STGCN	Centralized	32.74	52.16	45.15	73.50
	GWNet	Centralized	30.46	46.77	42.25	69.18
	FedGRU	Federated	43.54	69.19	61.95	94.52
	FASTGNN	Federated	42.31	66.49	59.36	92.18
	FedSTN	Federated	41.18	65.24	58.87	92.07
	SFL	Federated	40.00	64.31	57.19	91.92
	CNFGNN	Federated	38.22	61.59	52.87	88.65
	FedCroST (ours)	Federated	**35.74**	**57.19**	**48.70**	**84.91**

4.2 Performance Comparison

The average performances of our approach and all alternative baselines on the two datasets are summarized in Table 3. We compute the averaged MAE and RMSE of each model. From the experimental reports, several observations can be drawn as follows:

(i) The performance of the separated approaches is the worst. On the one hand, it cannot be trained centrally to capture the inherent complex spatio-temporal correlations among nodes, and on the other hand, it is unable to share global topological relationships and propagate similar semantics among local features in a federated aggregation manner. Therefore, isolated local training at each node cannot meet the landing requirements of spatio-temporal mobility modeling.

(ii) The centralized methods achieve excellent performance due to the powerful spatial correlation modeling capabilities of GNNs, especially with separated approaches that can solely capture temporal dependencies in the data. However, in data silo scenarios, the difficulty of data sharing makes centralized training impractical.

(iii) In the federated methods, personalized federated learning methods consistently outperform their counterparts. Notably, the performance of FedCroST surpasses the others by a large margin on both datasets. In addition, when compared to certain centralized methods, our approach maintains competitive performance levels.

The notable improvement of our proposed FedCroST can be attributed to the effective implementation of model

personalization by exploring distribution characteristics in spatio-temporal data, and we will further investigate our conjecture in the following parts.

4.3 Ablation Study

To further illustrate the effectiveness of different components in FedCroST, we design five variants to conduct ablation experiments and analyze experimental results on both KnowAir and Covid-TX datasets, including: (i) FedCroST without Temporal Prompt Representation (**w/o TPR**), (ii) FedCroST without Spatial Prompt Representation (**w/o SPR**), (iii) FedCroST without spatio-temporal Prompts (**w/o Prompt**), (iv) FedCroST without Denoising AutoEncoder (**w/o DAE**), (v) Replace diffusion module with Temporal Convolutional Network (**TPR-TCN**), (vi) Replace the Adaptive Spatial Structure Partition module with Spectral Clustering (**SPR-SC**), (vii) Replace the Linear Attention module with Feature Concatenation (**SPR-FC**), (viii) Replace the conditional diffusion module with the vanilla diffusion module (**TPR-Diff**).

TABLE 4
Component ablation analysis of FedCroST on 12-step ahead forecasting (RMSE).

Methods	KnowAir		Covid-TX	
	MAE	RMSE	MAE	RMSE
w/o TPR	30.07	38.72	51.84	87.13
w/o SPR	30.01	38.40	50.71	86.54
w/o Prompt	30.65	39.09	54.12	88.37
w/o DAE	29.75	38.34	49.93	85.99
TPR-TCN	29.95	38.37	50.52	86.38
SPR-SC	29.73	38.29	49.98	86.01
SPR-FC	29.61	38.22	49.65	86.29
TPR-Diff	29.88	38.33	50.26	86.35
FedCroST (ours)	**29.28**	**38.15**	**48.70**	**84.91**

As illustrated in Table 4, the removal of the denoising autoencoder on the client side harms performance to some extent, which suggests the presence of implicit interconnections among multi-view spatio-temporal features, while simple feature concatenation cannot effectively stimulate the complementary effects of multi-view spatio-temporal features. Substituting the diffusion module with TCN severely results in a discernible performance deterioration, which demonstrates that the diffusion module generates a high-quality representation of the temporal distribution through the denoising process, whereas the TCN is unable to capture the long-term temporal patterns limited by the size of the receptive field. The elimination of temporal prompts degrades the performance dramatically, which indicates that the historical time series highly hints at the temporal trends in future states. Removing the spatial prompts also immensely impairs the performance, which proves that a subgraph composed of nodes with similar patterns provides insights into the evolution pattern of the nodes within it. Simultaneous removal of both prompt modules achieves the worst performance, reinforcing the conclusion that spatio-temporal prompts can guide the local model training and enhance pattern mining of data distribution for clients. Using spectral clustering to replace our proposed adaptive spatial structure partitioning impairs the model performance to a large extent, suggesting that end-to-end adaptive spatial structure partitioning is important for the identification of spatial latent distributions and that simple preprocessing methods cannot handle complex spatial correlations effectively. The use of the vanilla diffusion model severely impairs the model performance, which confirms that the long-term history state is indispensable for understanding the temporal distribution, and our proposed conditional diffusion model is an effective solution to enhance the personalized temporal information in the local model. In multiscale spatial modeling, simply concatenating multi-granularity spatial features deteriorates the model performance to a certain extent, which suggests that the introduction of the attention mechanism can adaptively perceive neighborhood topological relationships, allowing beneficial multi-granularity spatial patterns to facilitate the learning of personalized spatial information.

4.4 Exploratory Study

4.4.1 Generalization Study

Many of the studies in the realm of federated spatio-temporal learning are motivated by intelligent transportation systems, commonly denoted as Road Side Units (RSU), wherein each unit exhibits a distinctive distribution of traffic patterns within the same city. To affirm the generalizability of our approach across diverse spatio-temporal mobility modeling tasks, we deliberately select supplementary traffic datasets (namely METR-LA and PEMS-BAY) for experimental validation, with detailed outcomes presented in Table 5. Here, METR-LA and PEMS-BAY are traffic flow datasets held by 207 loop sensors on the highway of Los Angeles County and 325 sensors in the Bay Area respectively. It is noteworthy that both METR-LA and PEMS-BAY datasets are also empirically adopted in CNFGNN [30], which further substantiates the impartiality of our experiments and fortifies their persuasiveness.

TABLE 5
Comparison results on traffic flow forecasting task.

(a) Comparison results on **METR-LA** dataset.

Methods	3-step ahead		6-step ahead		12-step ahead	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
FedGRU	5.32	7.76	6.39	9.64	7.45	11.53
SFL	4.52	6.84	5.61	9.07	6.88	10.96
CNFGNN	4.48	6.53	5.33	8.58	6.55	10.21
FedCroST (ours)	**4.23**	**5.22**	**4.22**	**6.85**	**5.01**	**8.57**

(b) Comparison results on **PEMS-BAY** dataset.

Methods	3-step ahead		6-step ahead		12-step ahead	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
FedGRU	2.51	4.09	2.74	5.17	3.74	6.93
SFL	2.32	4.30	2.98	5.74	3.59	6.67
CNFGNN	2.09	3.86	2.61	5.27	3.25	6.38
FedCroST (ours)	**1.90**	**3.52**	**2.39**	**4.87**	**3.00**	**5.94**

According to the experimental results in Table 5, we can observe that the performance of FedCroST surpasses the others by a large margin across both datasets, which also proves the considerable generalizability of FedCroST on different datasets and tasks.

4.4.2 Robustness Study

Impact of the historical time step. To delve deeper into the impact of the historical time step P on the prediction results, we varied P values to 6, 12, and 24 to observe changes in the prediction performance of our model and the baselines.

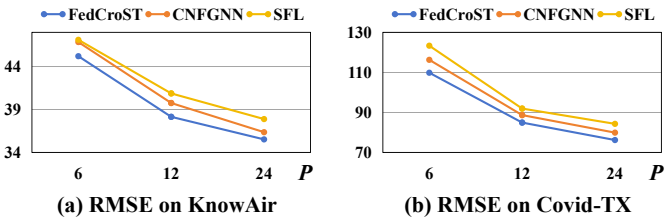

Fig. 6. RMSE of alternative methods under different time step P.

The experimental results are depicted in Fig. 6, where the model's performance improves gradually with increasing history steps, consistently outperforming baselines in each setting. However, due to the limitations of the model's learning capability, the improvement in results for all methods when increasing the time step from 12 to 24 is less pronounced compared to previous increments. This indicates that performance does not increase linearly with the time step, and there is an upper limit to performance improvement. At the same time, we intuitively recognize that increasing the time step not only demands a larger data size but also incurs greater computational overhead. Therefore, after considering these factors comprehensively, we chose $P = 12$ as a balanced compromise to achieve relatively optimal performance.

Impact of the data silo topologies. To further validate the effectiveness of our model under different data silo topologies, we randomly perturb the topology of each dataset by randomly masking 5% and 10% of the edges between nodes, to observe the changes in the prediction performance of our model and the baselines.

TABLE 6
RMSE of alternative methods under different data silo topologies.

Masking ratio	0%	5%	10%
SFL	40.88	42.81	44.69
CNFGNN	39.76	41.38	43.20
FedCroST (ours)	38.15	39.65	40.78

The experimental results on KnowAir, as shown in Table 6, demonstrate that our method outperforms the baselines under various topological changes. Moreover, after masking a certain number of edges, the difficulty of modeling the spatial information of the topology structure increases, but our method still maintains considerable performance. Therefore, our model has excellent spatial information extraction capability and can effectively handle spatio-temporal mobility modeling under diversified data silo topologies.

4.4.3 Scalability Study

To explore the scalability of our proposal, we incorporate alternative aggregation algorithms into FedCroST, including FedProx [33], MOON [34] and FedPer [35], which are the representative federated learning methods. Specifically, we substitute the FedAvg utilized in Section 3.5 with the aforementioned methods, and subsequently compare the performance of the integrated methods against their direct application. The results of 12-step forecasting experiments are summarized in Table 7. According to the results, our proposal can also enhance the performance of other federated learning algorithms, which confirms that the proposed FedCroST has good scalability. In contrast to other methods, our approach adeptly leverages the data distribution space, providing an orthogonal and complementary perspective for personalized implementations of the model-centric federated algorithms.

TABLE 7
Scalability analysis of FedCroST. (*"+CroST"* denotes the algorithm integrated with FedCroST.)

Methods	Integration	KnowAir	Covid-TX
FedProx	×	40.28	92.25
FedProx+*CroST*	✓	**38.26**	**85.33**
MOON	×	43.57	97.25
MOON+*CroST*	✓	**40.95**	**89.81**
FedPer	×	42.36	94.68
FedPer+*CroST*	✓	**39.07**	**87.10**

4.4.4 Computational & Communication Cost Study

To demonstrate the practicality of our proposal, we conduct comparative experiments on computational cost and communication cost. The computational cost of models on clients is quantified in terms of floating-point operations (FLOPS), while the total size of parameters transmitted during the training stage (Train Communication Cost) until the model attains its lowest validation error is presented. As depicted in 8, the computational and communication costs of our method are moderate and deemed acceptable. To mitigate the communication cost during the training stage, we adopt an alternating optimization method following [30] to alternately train the client- and server-side models, i.e., optimize client-side parameters when fixing server-side parameters and vice versa, which significantly reduces the volume of parameters transmitted. Considering the significant performance improvement demonstrated in Table 3, the costs of our method are deserved and meet the need for practicability.

TABLE 8
Comparison of the computation cost and the communication cost of FedCroST and baselines.

Methods	Computational cost (GFLOPS)	Train Communication Cost (GB)	
		KnowAir	Covid-TX
FedGRU	0.15	74.58	148.59
CNFGNN	0.15	141.78	288.41
SFL	0.36	163.55	334.42
FedCroST	**0.18**	**149.88**	**297.25**

4.4.5 Hyperparameter Study

To assess the impact of distinct parameter configurations, we perform experiments to evaluate the performance

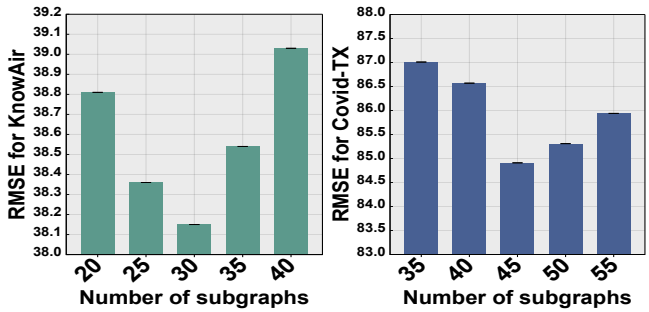

Fig. 7. Hyperparameter analysis of FedCroST (RMSE).

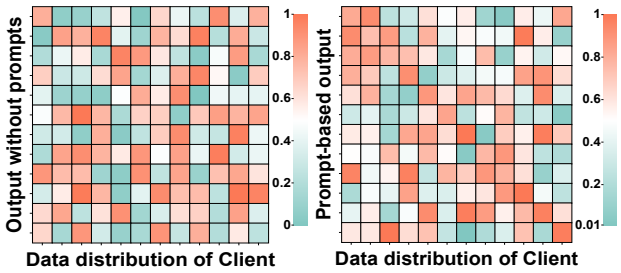

Fig. 8. Heatmap of similarity between the decoder's representation and actual data distribution.

of FedCroST with different configurations of the number of subgraphs N^s. Specifically, we respectively vary within the ranges within the ranges $\{20, 25, 30, 35, 40\}$ and $\{35, 40, 45, 50, 55\}$ while maintaining default values for other parameters, and then scrutinize the resultant fluctuations in model performance. The performance variations of 12-step ahead prediction are characterized in Fig. 7. According to the results, we obtain the optimal configurations of $N^s = \{30, 45\}$ for KnowAir and Covid-TX datasets, respectively. Notably, our findings reveal that the performance of the model is sensitive to the number of subgraphs on different datasets, signifying the data-specific nature of patterns within the subgraph structure. In essence, spatial prompts embody data-specific information that manifests differently across distinct data distributions.

4.4.6 *Visual Analysis*

To further intuitively validate the efficacy of spatio-temporal prompts, we visualize the heatmap of similarity between the decoder's representation and actual data distribution. Two distinct representations are presented, one incorporating spatio-temporal prompts and the other without, allowing for a comparative analysis of their similarities to the original data distribution. The horizontal and vertical coordinates in Fig. 7 represent the actual distribution and the representation output of the clients. Since our goal is to obtain representations that approximate the actual distribution, if high similarity values are distributed along the diagonal line, it indicates that the actual distribution and the representation output of the same client are generally consistent. As depicted in Fig. 8, in contrast to the left subfigure, where the similarity correspondence between the representations and the actual distribution is not significant, the high similarity values in the right subfigure are mainly distributed along the diagonal, This indicates that the prompt-based representations in most clients show high consistency with the actual data distribution. A higher degree of consistency between the representation utilizing spatio-temporal prompts and the actual data distribution signifies the accurate delineation of the data characteristics, consequently enabling the models to more effectively capture local data patterns and facilitating model personalization.

5 Discussion

5.1 Connection with Privacy Protection

While our work targets to solve the problem of spatio-temporal learning across data silos rather than privacy protection, the latter remains an intriguing aspect within FL system. In our framework, we ensure that the data is exclusively stored in the local client, the data between the clients is isolated from each other, and only the gradient rather than the raw data is uploaded to the server during training, guaranteeing that the data is not available to any other party. Potential privacy concerns arising from gradient leakage may be a consideration for readers. However, such concerns are mitigated within our framework. Each client holds its own local module, and only part of the gradient needs to be uploaded, while the predictor is not involved in the parameter updating process between clients and the server. This means that it is hard to reverse-engineer the original data using gradients, as not all gradients are exposed. Furthermore, we include coefficients in the local loss function, which makes it difficult for an attacker to perform inference attacks to infer the original data.

5.2 Limitation

A prevalent limitation in the domain of federated spatio-temporal learning, as highlighted in the literature [3], is the absence of realistic cross-silo spatial-temporal datasets suitable for benchmarking purposes. Existing approaches within this field commonly rely on evaluations using artificially partitioned spatio-temporal data. Nevertheless, the sustained advancement of this field necessitates the availability of realistic federated datasets to support experimental assessments aligned with practical applications. Due to the inherent difficulty in constructing datasets, existing work, including all the representative baselines, is almost always evaluated on public datasets, which inevitably leads to a gap with real data silo scenarios. To evaluate our proposal, we artificially divide the public datasets, which remain a simulation of real data silo scenarios and are as close as possible to the actual application situation.

5.3 Future Work

i) Integration of Advanced Privacy Technologies into FedCroST: While our primary focus lies in addressing data silo scenarios, federated learning is also considered for addressing the privacy concerns, such as cross-device federated applications [71], [72]. In future endeavors, we intend to incorporate differential privacy to mitigate potential gradient leakage during communication, thereby eliminating the risk of data recovery. Additionally, we aim to fortify the framework against backdoor attacks, enhancing its robustness and contributing to the evolving landscape of secure mobile computing.

ii) Establishment of a Federated Spatio-Temporal Learning Benchmark: A promising avenue for future exploration involves determining optimal data partitioning strategies across silos [3], which lays the groundwork for the development of a comprehensive cross-silo federated benchmark. Such a benchmark will play a pivotal role in advancing the evaluation and comparison of federated spatio-temporal learning methods in diverse real-world scenarios.

6 Related Work

We briefly review related literature, and then distinguish our proposal from existing work.

6.1 Spatio-temporal Mobility Modeling

Spatial-temporal mobility modeling has played an important role in many mobile computing applications in the past decades [1], [5], [6], [10] and witnessed unprecedented advancements in data-driven solutions, such as deep learning-based approaches [18].

Recurrent Neural Network-based models, exemplified by LSTM [73] and GRU [70], have demonstrated proficiency in extracting patterns from sequential data and have been effectively employed in spatio-temporal forecasting applications [74]. However, spatial dependencies in the spatio-temporal data were omitted which encourages researchers to propose methods that explicitly account for spatial relationships. Convolutional Neural Networks (CNNs) have been extensively utilized to capture the spatial correlations in grid-based mobility networks [75]. Nevertheless, a significant portion of spatio-temporal data exhibits a graph-structured nature, as observed in cellular networks [76] and road networks [77], rendering CNNs unsuitable for effectively representing spatial features in such contexts.

To incorporate spatial dependencies more adeptly, Graph Neural Networks (GNNs) are introduced to spatio-temporal mobility modeling and have achieved great success underpinned by their inherent ability to discern complex relationships and dependencies within graph-structured data [20], [21], [22]. Specifically, STGCN [20] leverages ChebNet graph convolution and 1D convolution to extract spatial dependencies and temporal correlations in traffic data. Graph WaveNet [24] captures the spatial correlation with a diffusion convolution layer and learns the temporal correlation using generic TCN. ASTGCN [21] uses attention-based spatial-temporal graph convolutions to model dynamic spatial-temporal features of traffic flows. AGCRN [22] introduces node adaptive parameter learning to automatically capture node-specific spatial and temporal correlations in time-series data without a predefined graph. MVSTGN [76] investigates diverse spatial-temporal characteristics in the cellular traffic network from multiple views and combines attention and convolution mechanisms for traffic pattern analysis. MGAT [78] is a cross-city traffic prediction framework that extracts all the multi-granular regional features of multiple source cities to maximize the preservation of useful information. DMSTG [17] fuses features from multiple views into their forecasting framework to capture complex spatial-temporal dependencies. Notwithstanding the superiority exhibited by the aforementioned methods, it is imperative to note their common reliance on large-scale datasets centrally collected, which inevitably puts forward strict requirements for the centralized management of data [25]. In the contemporary landscape characterized by an escalating emphasis on data protection, data is frequently held by various distributed entities (e.g., companies or organizations), which are isolated from each other and confront limitations on data sharing [26], [27], [28]. The data separately maintained by each entity is typically insufficient for effective spatio-temporal mobility modeling thus hindering the development of downstream applications.

Consequently, there arises a pressing need to delve into the techniques facilitating cross-silo spatio-temporal mobility modeling, aligning with the imperatives of the burgeoning mobile big data era, which is exactly what we address in this paper.

6.2 Federated Learning in Spatio-temporal Mobility Modeling across Data Silos

To address the increasingly prevalent data silo scenario, there have been some attempts to introduce Federated Learning (FL) into spatiotemporal learning, which has emerged as the predominant paradigm for achieving distributed computing without sharing data [79], [80].

In particular, FedGRU [25] integrates emerging FL with a GRU neural network for spatiotemporal learning, which updates universal learning models through a parameter aggregation mechanism rather than directly sharing raw data among organizations. Recognizing the graph structure inherent in spatio-temporal data, FASTGNN [26] amalgamates a GNN-based model for local training and a novel FL strategy to protect the shared topological information. FedSTN [29] proposes a federated deep learning based on the spatial-temporal long and short-term networks to predict traffic flow by utilizing observed historical traffic data. CNFGNN [30] aggregates local temporal embeddings uploaded from clients and employs GNNs to obtain spatial embeddings, which are sent back to the corresponding clients for forecasting. Nevertheless, as depicted in Fig. 1, the distribution pattern of spatio-temporal data across different data silos is inconsistent, presenting a pervasive challenge termed "distribution heterogeneity" in the field of spatio-temporal mobility modeling [1], [18]. Consequently, the conventional federated learning paradigm, as employed in the aforementioned approaches to construct a unified global model for all clients, proves ineffective in addressing heterogeneous spatio-temporal patterns present in each client, which severely undermines the model's performance in local tasks.

This problem is turned around by the proposal of personalized federated learning, which aims to furnish each client with a dedicated model tailored to better fit its local data. For instance, GOF-TTE [31] proposes an online personalized federated learning framework to fill the gap between personal privacy and model performance by dynamically updating the global traffic state. Structured federated learning (SFL) framework [32] learns both the global and personalized models simultaneously using client-wise relation graphs and clients' private data. While these approaches draw inspiration from the popular ideas of personalized

federated learning (e.g., regularizing model parameters [33], [34] or adding extra personalized layers for the client-side model [35]), they fall short in contributing to personalization from the perspective of the inherent spatio-temporal nature of the data, specifically addressing the pervasive challenge of distribution heterogeneity in the field of spatio-temporal mobility modeling [1], [18].

In fact, the challenge of federated learning in handling spatio-temporal data is fundamentally caused by distribution heterogeneity. Therefore, our FedCroST is proposed to extract heterogeneous spatio-temporal patterns among clients from the vantage point of spatio-temporal distribution, providing a novel and insightful avenue for enhancing spatio-temporal mobility modeling across silos.

7 Conclusion

In this paper, we propose a spatio-temporal distribution-oriented personalized federated learning framework (FedCroST) for cross-silo spatio-temporal mobility modeling by leveraging the intrinsic properties of spatio-temporal data. To uncover potential characteristics within temporal distribution, we devise a hidden state-based conditional diffusion module to generate temporal prompts rich in the evolution within time series. To capture the structure distribution inherent in node neighborhoods, we propose adaptive spatial structure partition, and accordingly learn spatial prompts that augment the spatial information representation. Furthermore, to effectively harness the learned multi-view spatio-temporal features, we introduce a denoising autoencoder to generate personalized ST representations for local tasks. Extensive experiments on real-world datasets in spatio-temporal mobility modeling demonstrate the superiority of our approach. Therefore, our work innovatively highlights the significance of latent spatio-temporal data distributions in enabling personalized federated spatio-temporal learning, which provides new insights into modeling spatio-temporal mobility in the often-overlooked yet increasingly prevalent data silo scenario.

References

[1] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–41, 2018.

[2] E. Wang, M. Zhang, B. Yang, Y. Yang, and J. Wu, "Large-scale spatiotemporal fracture data completion in sparse crowdsensing," *IEEE Transactions on Mobile Computing*, 2023.

[3] Y. Belal, S. B. Mokhtar, H. Haddadi, J. Wang, and A. Mashhadi, "Survey of federated learning models for spatial-temporal mobility applications," *arXiv preprint arXiv:2305.05257*, 2023.

[4] L. You, M. Danaf, F. Zhao, J. Guan, C. L. Azevedo, B. Atasoy, and M. Ben-Akiva, "A federated platform enabling a systematic collaboration among devices, data and functions for smart mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 4060–4074, 2023.

[5] L. Beck and H. Schuldt, "City-stories: A spatio-temporal mobile multimedia search system," in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 193–196.

[6] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2190–2202, 2018.

[7] Y. Zhang, B. Wang, Z. Shan, Z. Zhou, and Y. Wang, "Cmt-net: A mutual transition aware framework for taxicab pick-ups and drop-offs co-prediction," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1406–1414.

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

[9] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[10] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[11] Y. Liang, Y. Xia, S. Ke, Y. Wang, Q. Wen, J. Zhang, Y. Zheng, and R. Zimmermann, "Airformer: Predicting nationwide air quality in china with transformers," *arXiv preprint arXiv:2211.15979*, 2022.

[12] Y. Ma, P. Gerard, Y. Tian, Z. Guo, and N. V. Chawla, "Hierarchical spatio-temporal graph neural networks for pandemic forecasting," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1481–1490.

[13] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 1655–1661.

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

[15] S. N. SM, P. R. Yasa, M. Narayana, S. Khadirnaikar, and P. Rani, "Mobile monitoring of air pollution using low cost sensors to visualize spatio-temporal variation of pollutants at urban hotspots," *Sustainable Cities and Society*, vol. 44, pp. 520–535, 2019.

[16] L. You, R. Zhu, M.-P. Kwan, M. Chen, F. Zhang, B. Yang, M. S. Wong, and Z. Qin, "Unraveling adaptive changes in electric vehicle charging behavior toward the postpandemic era by federated meta-learning," *The Innovation*, vol. 5, no. 2, 2024.

[17] Z. Diao, X. Wang, D. Zhang, G. Xie, J. Chen, C. Pei, X. Meng, K. Xie, and G. Zhang, "Dmstg: Dynamic multiview spatio-temporal networks for traffic forecasting," *IEEE Transactions on Mobile Computing*, 2023.

[18] S. Wang, J. Cao, and P. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE transactions on knowledge and data engineering*, 2020.

[19] Y. Zhang, P. Wang, B. Wang, X. Wang, Z. Zhao, Z. Zhou, L. Bai, and Y. Wang, "Adaptive and interactive multi-level spatio-temporal network for traffic forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[20] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[21] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[22] 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.

[23] B. Wang, P. Wang, Y. Zhang, X. Wang, Z. Zhou, L. Bai, and Y. Wang, "Towards dynamic spatial-temporal graph learning: A decoupled perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8, 2024, pp. 9089–9097.

[24] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 1907–1913.

[25] Y. Liu, J. James, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, 2020.

[26] C. Zhang, S. Zhang, J. James, and S. Yu, "Fastgnn: A topological information protected federated learning approach for traffic speed forecasting," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8464–8474, 2021.

[27] D.-V. Nguyen and K. Zettsu, "Spatially-distributed federated learning of convolutional recurrent neural networks for air pollution prediction," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 3601–3608.

[28] N. Zhang, Q. Ma, and X. Chen, "Enabling long-term cooperation

in cross-silo federated learning: A repeated game perspective," *IEEE Transactions on Mobile Computing*, 2022.

[29] X. Yuan, J. Chen, J. Yang, N. Zhang, T. Yang, T. Han, and A. Taherkordi, "Fedstn: Graph representation driven federated learning for edge computing enabled urban traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[30] C. Meng, S. Rambhatla, and Y. Liu, "Cross-node federated graph neural network for spatio-temporal data modeling," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1202–1211.

[31] Z. Zhang, H. Wang, Z. Fan, J. Chen, X. Song, and R. Shibasaki, "Gof-tte: Generative online federated learning framework for travel time estimation," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24 107–24 121, 2022.

[32] F. Chen, G. Long, Z. Wu, T. Zhou, and J. Jiang, "Personalized federated learning with a graph," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2022, pp. 2575–2582, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2022/357

[33] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[34] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 713–10 722.

[35] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[36] 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.

[37] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, "Graph convolutional networks for road networks," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019, pp. 460–463.

[38] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1695–1704.

[39] J. Wang, N. Wu, W. X. Zhao, F. Peng, and X. Lin, "Empowering a* search algorithms with neural networks for personalized route recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 539–547.

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

[41] E. Wang, W. Liu, W. Liu, Y. Yang, B. Yang, and J. Wu, "Spatiotemporal urban inference and prediction in sparse mobile crowdsensing: A graph neural network approach," *IEEE Transactions on Mobile Computing*, 2022.

[42] C. Zhang, S. Zhang, X. Zou, S. Yu, and J. James, "Towards large-scale graph-based traffic forecasting: A data-driven network partitioning approach," *IEEE Internet of Things Journal*, 2022.

[43] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[44] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[45] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–39, 2023.

[46] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csdi: Conditional score-based diffusion models for probabilistic time series imputation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 804–24 816, 2021.

[47] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8857–8868.

[48] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

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

[50] Z. Zhou, Y. Wang, X. Xie, L. Chen, and C. Zhu, "Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[51] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *Advances in neural information processing systems*, vol. 31, 2018.

[52] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *International conference on machine learning*. PMLR, 2020, pp. 874–883.

[53] E. Ranjan, S. Sanyal, and P. Talukdar, "Asap: Adaptive structure aware pooling for learning hierarchical graph representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5470–5477.

[54] Z. Zhang, J. Bu, M. Ester, J. Zhang, Z. Li, C. Yao, H. Dai, Z. Yu, and C. Wang, "Hierarchical multi-view graph pooling with structure learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 545–559, 2021.

[55] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[57] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[58] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International conference on machine learning*. PMLR, 2020, pp. 5156–5165.

[59] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[60] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[61] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[62] S. Wu, M. Zhang, Y. Li, C. Yang, and P. Li, "Graph federated learning with hidden representation sharing," *arXiv preprint arXiv:2212.12158*, 2022.

[63] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 394–21 405, 2020.

[64] S. Wang, Y. Li, J. Zhang, Q. Meng, L. Meng, and F. Gao, "Pm2.5-gnn: A domain knowledge enhanced graph neural network for pm2.5 forecasting," in *Proceedings of the 28th international conference on advances in geographic information systems*, 2020, pp. 163–166.

[65] Y. Chen, I. Segovia-Dominguez, B. Coskunuzer, and Y. Gel, "Tamps2gcnets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting," in *International Conference on Learning Representations*, 2022.

[66] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[67] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8485–8493.

[68] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[69] B. Pan, U. Demiryurek, and C. Shahabi, "Utilizing real-world transportation data for accurate traffic prediction," in *2012 ieee 12th international conference on data mining*. IEEE, 2012, pp. 595–604.

[70] A. F. M. Agarap, "A neural network architecture combining gated recurrent unit (gru) and support vector machine (svm) for intrusion detection in network traffic data," in *Proceedings of the 2018 10th international conference on machine learning and computing*, 2018, pp. 26–30.
[71] S. P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S. Reddi, S. U. Stich, and A. T. Suresh, "Breaking the centralized barrier for cross-device federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 663–28 676, 2021.
[72] M. H. ur Rehman, A. M. Dirir, K. Salah, E. Damiani, and D. Svetinovic, "Trustfed: A framework for fair and trustworthy cross-device federated learning in iiot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8485–8494, 2021.
[73] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values," *Transportation Research Part C: Emerging Technologies*, vol. 118, p. 102674, 2020.
[74] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.
[75] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.
[76] Y. Yao, B. Gu, Z. Su, and M. Guizani, "Mvstgn: A multi-view spatial-temporal graph network for cellular traffic prediction," *IEEE Transactions on Mobile Computing*, 2021.
[77] L. Yan, H. Shen, J. Zhao, C. Xu, F. Luo, and C. Qiu, "Catcharger: Deploying wireless charging lanes in a metropolitan road network through categorization and clustering of vehicle traffic," in *IEEE INFOCOM 2017-IEEE conference on computer communications*. IEEE, 2017, pp. 1–9.
[78] J. Mo and Z. Gong, "Cross-city multi-granular adaptive transfer learning for traffic flow prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11 246–11 258, 2022.
[79] L. You, Q. Chen, H. Qu, R. Zhu, J. Yan, P. Santi, and C. Ratti, "Fmgcn: Federated meta learning-augmented graph convolutional network for ev charging demand forecasting," *IEEE Internet of Things Journal*, 2024.
[80] S. Liu, L. You, R. Zhu, B. Liu, R. Liu, H. Yu, and C. Yuen, "Afm3d: An asynchronous federated meta-learning framework for driver distraction detection," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

Yudong Zhang (Student Member, IEEE) is now a Ph.D. candidate in the School of Artificial Intelligence and Data Science, University of Science and Technology of China (USTC). He received his bachelor's degree from the University of Electronic Science and Technology of China (UESTC) in 2020. He has published over twenty research papers in top journals and conferences such as IEEE TPAMI, IEEE TITS, IEEE TVT, ICLR, SIGKDD, AAAI, WSDM, and ICDM. His current research interests include mobile data mining and urban computing.

Xu Wang is now an Associate Researcher at the University of Science and Technology of China (USTC). He received his Ph.D. degree at USTC in 2023, under the supervision of Professor Zheng-Jun Zha and Yang Wang. He got his bachelor's degree in automation at North Eastern University in 2017. His research interests mainly encompass spatio-temporal data mining, time series analysis, and the application of AI in scientific research.

Pengkun Wang (Member, IEEE) is now an Associate Researcher at the University of Science and Technology of China (USTC). He got his Ph.D. degree at USTC in 2023, under the supervision of Professor Qi Liu and Yang Wang. His research interest mainly includes open environment machine learning, spatio-temporal data mining, and generalized AI for Science.

Binwu Wang received his Ph.D. degree at the School of Data Science, University of Science and Technology of China (USTC) in 2024. He has published over ten papers in top conferences and journals such as IEEE TMC, IEEE TITS, IEEE TVT, ICLR, SIGKDD, IJCAI, DASFAA, and WSDM. His main research interests include mobile data mining and continual learning, especially their applications in urban computing.

Zhengyang Zhou (Member, IEEE) is now an Associate Researcher at Suzhou Institute for Advanced Research, University of Science and Technology of China (USTC). He got his Ph.D. degree at USTC in 2023. He has published over twenty papers in top conferences and journals such as IEEE TMC, IEEE TKDE, KDD, ICLR, WWW, AAAI, NeurIPS, and ICDE. His main research interests include human-centered urban computing and mobile data mining.

Yang Wang (Senior Member, IEEE) is now an Associate Professor at the School of Computer Science and Technology, School of Software Engineering, and School of Artificial Intelligence and Data Science at the University of Science and Technology of China (USTC). He got his Ph.D. degree at USTC in 2007. Since then, he keeps working at USTC till now as a postdoc and an associate professor successively. Meanwhile, he also serves as the vice dean of the School of Software Engineering of USTC. His research interests mainly include wireless (sensor) networks, distributed systems, data mining, and machine learning, and he is also interested in all kinds of applications of AI and data mining technologies, especially in urban computing and AI4Science. His work has been published in top-tier conferences and journals like ICLR, NeurIPS, ICML, KDD, AAAI, WWW, IEEE TKDE, and IEEE TPAMI, with over fifty papers as the first author or corresponding author. Additionally, he also serves as a reviewer for leading conferences and journals including ICLR, KDD, NeurIPS, ICML, and IEEE TKDE.