Collaborative Imputation of Urban Time Series through Cross-city Meta-learning

Tong Nie, Wei Ma† *Member, IEEE*, Jian Sun†, Yu Yang, Jiannong Cao *Fellow, IEEE*

Abstract—Urban time series, such as mobility flows, energy consumption, and pollution records, encapsulate complex urban dynamics and structures. However, data collection in each city is impeded by technical challenges such as budget limitations and sensor failures, necessitating effective data imputation techniques that can enhance data quality and reliability. Existing imputation models, categorized into learning-based and analytics-based paradigms, grapple with the trade-off between capacity and generalizability. Collaborative learning to reconstruct data across multiple cities holds the promise of breaking this trade-off. Nevertheless, urban data's inherent irregularity and heterogeneity issues exacerbate challenges of knowledge sharing and collaboration across cities. To address these limitations, we propose a novel collaborative imputation paradigm leveraging meta-learned implicit neural representations (INRs). INRs offer a continuous mapping from domain coordinates to target values, integrating the strengths of both paradigms. By imposing embedding theory, we first employ continuous parameterization to handle irregularity and reconstruct the dynamical system. We then introduce a cross-city collaborative learning scheme through model-agnostic meta learning, incorporating hierarchical modulation and normalization techniques to accommodate multiscale representations and reduce variance in response to heterogeneity. Extensive experiments on a diverse urban dataset from 20 global cities demonstrate our model's superior imputation performance and generalizability, underscoring the effectiveness of collaborative imputation in resource-constrained settings.

Index Terms—Time Series Imputation, Implicit Neural Representations, Cross-city Generalization, Meta Learning.

1 Introduction

Time series measured in urban agglomerations, such as mobility flows, energy consumption, and pollution records, represent time-dependent urban patterns and dynamics. These high-granular quantities can be exploited by data-driven models to reflect latent profiles of cities, such as human activity, socio-economic and welfare [1]. To utilize such data, either Eulerian sensors with dense spatial deployments or Lagrangian sensors with high temporal coverage are desired to measure them [2]. However, access to city-scale holographic data is far from easy due to factors such as installation and maintenance costs, adverse observation conditions, and system errors, hindering the usage of urban computing applications [3], [4]. Therefore, data imputation technique has emerged to compensate for the lack of full observations and enhance data quality and reliability.

Existing data imputation models generally fall into two paradigms: learning-based and analytics-based solutions.

- *Tong Nie is with the Department of Traffic Engineering, Tongji University, Shanghai, China, and the Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, SAR, China (E-mail: tong.nie@connect.polyu.hk).*
- *Wei Ma is with the Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, SAR, China (E-mail: wei.w.ma@polyu.edu.hk).*
- *Jian Sun is with the Department of Traffic Engineering, Tongji University, Shanghai, China (E-mail: sunjian@tongji.edu.cn).*
- *Yu Yang is with the Centre for Learning, Teaching, and Technology, The Education University of Hong Kong, Hong Kong, SAR, China (E-mail: yangyy@eduhk.hk).*
- *Jiannong Cao is with the Research Institute for Artificial Intelligence of Things, Department of Computing, The Hong Kong Polytechnic University (E-mail: jiannong.cao@polyu.edu.hk).*
- *Corresponding authors: Jian Sun and Wei Ma.*

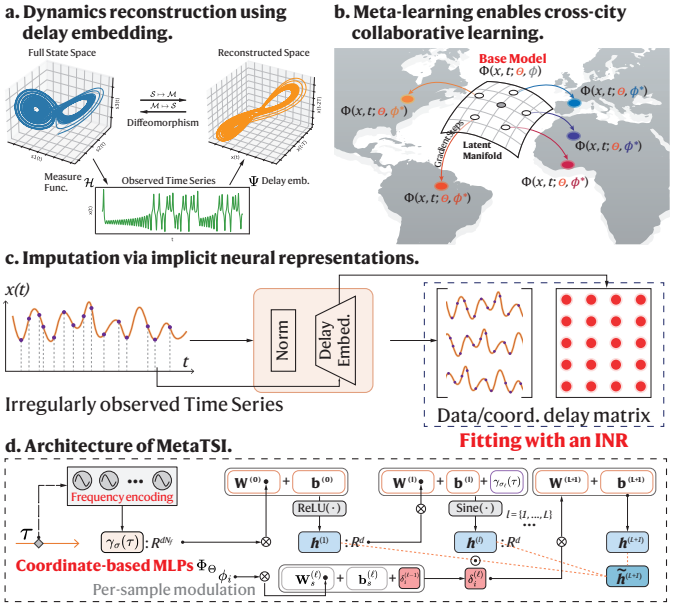


Fig. 1: Cross-city collaborative imputation framework.

However, both paradigms face *the dilemma of capacity and generalizability*. The former applies neural architectures that excel in fitting the time series distribution, such as RNNs [5], [6], probabilistic diffusion [7], [8], and Transformers [9], [10]. However, deep time series models struggle to comprehend underlying physical laws of data, leading to the potential for overfitting in masked training and lacking generalizability to unseen data outside of the training distribution [11]. The latter formalism designs analytical models to characterize commonly shared data properties, showing better general-

izability across data. One prominent example is the matrix completion model [12], [13], [14]. Unfortunately, they have limited model capacity and are restricted to fixed dimensions with a predefined input domain. As a result, they usually require per sample optimization, preventing them from modeling complex and diverse urban datasets.

Promisingly, the recent success of foundation models has demonstrated the potential of learning from the union of diverse datasets [15], [16]. Large models pretrained on cross-source data hold promise for breaking the trade-off. This inspires us to develop an expressive and generalizable imputation paradigm across multiple cities. However, public agencies in each city currently have to develop individual models based on separate expertise, which is resource-intensive with a specialized computational procedure. More critically, urban data within each city can only be used for the city and considered in isolation from each other. This hinders its accessibility and information share in resource-constrained contexts, where less developed cities have low observation rates and limited technical capacity to fully train a model. With a generalizable model and collaborative data governance, the shared knowledge from data-rich cities can inform different-but-related patterns in data-lacking cities.

Overall, the benefits of cross-city learning prompt us to explore the possibility of an innovative and alternative scheme for collaborative imputation. To this end, we summarize the challenges in urban time series imputation under practical constraints of observation conditions as: *irregularity and heterogeneity*. First, urban time series is irregularly sampled in nature. The data might be generated as a burst or with varying time intervals, and different sensors can have different sampling frequencies. Furthermore, missingness can manifest itself at arbitrary locations and timestamps [17]. To leverage state-of-the-art imputation models, one may convert an irregular time series into a regular time series. This can cause information loss and misinterpretation of the missing pattern. Second, urban data displays high heterogeneity [18], [19]. Measurements from different locations show diverse localized patterns such as multiple scales and frequencies. Consequently, collaborative learning from the joint distribution of heterogeneous data requires rigorous model design and large capacity. In addition, such heterogeneity increases the difficulty in discovering shared patterns that are generalizable between different cities.

In summary, most state-of-the-art models either apply instance-specific optimization with fixed spatial-temporal dimensions [13] or are trained to interpolate discrete signals in regular grids for a single data source [10]. They are biased toward particular data sources and compromise between accuracy and generality depending on the applied domain [20], hindering knowledge transfer across heterogeneous cities. To resolve this dilemma, we resort to an emerging approach called implicit neural representations (INRs). INRs have recently been shown to be proficient in learning representations from multimodal data, such as 2D images, 3D scenes, audios, time series, and spatiotemporal data [21], [22], [23], [24], [25], [26], [27]. INRs represent data instances by parameters of neural networks and establish continuous mappings from domain coordinate to quantity of interests. They inherit merits of both two paradigms with high expressivity and universal priors such as smoothness.

By capitalizing on the recent advancements in INRs, we first tackle the irregularity using continuous parameterization. Then, under the embedding theory [28], [29], the incomplete time series is imputed by reconstructing the dynamical system with global diffeomorphic embedding and a frequency-decomposed architecture. To achieve collaborative modeling of commonalities and shared patterns across cities to remove data barriers, we frame it as a generalizable representation learning task and convert the data of cities to centralized weights of neural networks and distribute them to different cities to complete imputation tasks. This process is accomplished by a model-agnostic meta learning (MAML) [30] framework in which an underlying low-dimensional manifold is discovered to learn the distribution of functions in a sensor-agnostic manner. To further deal with heterogeneity in cross-city transfer, masked instance normalization is proposed to alleviate the difficulty of learning with city-specific variance, and a hierarchical and multiscale modulation mechanism is developed to condition the model on heterogeneous patterns in a parameter-efficient way.

Integrating both physics and data priors into a holistic meta-learned INR architecture, we develop a novel cross-city collaborative imputation scheme that preserves both expressivity and generalizability. It enables the capture of shared knowledge across locations and efficient adaptation to new cities with few observations. To evaluate it, we collect a large-scale urban dataset from 20 cities around the globe, consisting of more than 8,000 time series with different resolutions, sampling frequencies, and spatiotemporal patterns. Experimental results suggest that our model contributes to the state-of-the-art with better imputation accuracy and generalizability. Our contributions are summarized as follows:

- We introduce a new time series imputation paradigm through INRs. By connecting the embedding theory with INRs, it reconstructs irregular urban time series with a frequency-decomposed deep architecture;
- A novel cross-city collaborative learning scheme is proposed via MAML, featuring a hierarchical modulation mechanism for multiscale conditioning and normalization to reduce instance-level heterogeneity;
- Empirical results show that our model outperforms SOTA baselines in both single-city and cross-city scenarios. The significance of the collaborative imputation scheme is also highlighted in few-shot settings.

The remainder is structured as follows. Sections 2 and 3 introduce the related work and background information. Section 4 introduces the collaborative imputation framework. Section 4.3 provides algorithmic analysis to interpret the model. Section 5 conducts experiments on a large-scale urban dataset. Finally, we conclude this study in Section 6.

2 Related Work

Spatiotemporal data imputation. As missing data is pervasive in spatiotemporal system, many recent studies have been devoted to establishing data imputation models [31]. Existing methods can be categorized into two types, i.e., learning-based and analytics-based methods. The former adopt deep neural networks to correlate observed values and learn to utilize the correlations to fill unobserved values.

Popular architectures such as recurrent neural networks, diffusion models, and Transformers are widely adopted [5], [6], [7], [8], [9], [10], [32], [33]. Analytical models approach the data imputation problem by solving an optimization problem associated with missing data patterns [13], [25], [34], [35], [36], [37], [38]. Representative methods include low-rank matrix factorization and tensor completion. The two existing paradigms facilitate the development of a specific imputation model for a particular dataset. Collaborative imputation of data across multiple datasets remains unexplored. In addition, although there are some imputation models designed for irregular time series [39], [40], [41], [42], their abilities in large-scale urban datasets are less explored.

Implicit neural representations. INRs are a class of methods to implicitly define a quantity of interest by associating the target value with its coordinate. By fitting the given data using coordinate-based neural networks, INRs provide an alternative to store and query data. Due to their continuous, expressive, efficient properties, INRs have achieved success in learning multimodal data, such as images, videos, 3D scenes, point cloud, and audio data [21], [22], [23], [24], [25], [43]. In addition, INRs have recently been adopted for spatiotemporal data [27], [44] and time series [26], [42], [45], with a main focus on reconstructing and forecasting tasks. Most of these studies either concentrate on a single task from a particular data source or evaluate the performances of INRs on various different tasks in parallel.

Meta-learning INRs. As an INR is fitted to a single instance using weights of neural networks, the learned weights cannot be applied to predict other instances directly. To address this limitation, generalizable implicit neural representations (GINRs) are developed using gradient- or Transformer-based meta-learning algorithms [46], [47], [48], [49]. The mechanism of GINRs is to model the distribution of functional representations using separate parameters, making them adaptable to different instances with a shared meta model and specific task models. Prior research on GINRs has focused on training them on homogeneous data, such as images, with the objective of developing a task-independent decoder. The potential of GINRs for cross-city learning of large-scale urban datasets has yet to be fully investigated.

Cross-city generalization. To encourage data and resource sharing between cities, previous work develops transfer-learning schemes for cross-city learning, with a particular focus on traffic forecasting problems [50], [51], [52], [53], [54], [55], [56]. With abundant observations available, the knowledge from source cities can be readily transferred to the target city. However, for the studied imputation task, cross-city generalization is more challenging with sparse data. In addition, as discrete neural networks are adopted as backbones, they can only deal with urban data with regular data organization, e.g., fixed spatial-temporal dimension.

Comparison with existing work. There are several studies that achieve cross-city generalization using meta-learning [57], [58], [59], [60], [61]. However, there are common drawbacks that limit their application. Specifically, MetaST [57] employs a discrete grid-based spatial-temporal network with a pattern memory module to distill periodicity, making it sensitive to irregular sensor distributions and unable to model continuous dynamics over arbitrary coordinates. ST-GFSL [58] generates node-level meta-parameters and uses a graph reconstruction loss to align heterogeneous urban topologies, but it relies on consistent graph structures and cannot capture fine-grained continuous variations in traffic flows. TPB [61] builds a discrete traffic pattern bank by clustering fixed-size patches, resulting in coarse-grained prototypes that struggle to adapt to irregular traffic behaviors in data-scarce cities. CrossTReS [59] learns to reweight source-city regions via a meta-learned network, but depends on region definitions that vary across cities, increasing the risk of negative transfer when region mappings are inconsistent. Note that the above models are designed for future forecasting with complete observations and are not directly applicable to sparse data reconstruction. Similar to the problem studied in this paper, Mest-GAN [60] employs a meta spatial-temporal generative adversarial network to generate traffic estimates without historical observations, yet adversarial training can be unstable when data is heterogeneous and lacks explicit mechanisms for handling cities with varying data dimensions. Collectively, these methods adopt either purely learning-based or discrete analytic components, struggling with the trade-off between model capacity and cross-city generalizability under irregular, sparse, and heterogeneous conditions. Instead, MetaTSI addresses these shortcomings by introducing meta-learned INRs to model time series as continuous functions over spatiotemporal coordinates, naturally accommodating irregular sampling and enabling smooth generalization across diverse domains. We further integrate techniques to capture multiscale urban dynamics and reduce variance arising from cross-city heterogeneity, achieving superior accuracy and robustness in resource-constrained urban environments.

3 Preliminary

Notations. We denote matrices by boldface capital letters e.g., $\boldsymbol{X} \in \mathbb{R}^{N\times T}$, vectors are denoted by boldface lowercase letters, e.g., $\boldsymbol{x} \in \mathbb{R}^{T}$, and scalars are lowercase letters, e.g., x. Without ambiguity, we also denote $\boldsymbol{x}(t) \in \mathbb{R}^{m}$ an arbitrary time series with time index t. The functional representation (or a mapping function) is abbreviated as $\Phi(\cdot)$. In the absence of remarks, calligraphic letters are used to denote the vector space, for example $\mathcal{X} \subseteq \mathbb{R}$.

Model-agnostic meta-learning. To utilize the shared pattern and knowledge between source and target data, a transfer learning algorithm needs to be established. The meta learning framework is developed for learning the learning algorithm itself. In particular, the model-agnostic meta-learning (MAML) algorithm [30] is designed to train a deep neural network on a variety of learning tasks, such that the source model is explicitly fine-tuned to generalize to a downstream task with few labels using a small number of gradient steps. MAML obeys the following iterative scheme:

$$\begin{aligned} &\underbrace{\min_{\Theta} \mathbb{E}_{\tau\sim p(\tau)} \left[\ell_{\tau}^{\text{test}}(\phi_{\tau,K}(\Theta))\right]}_{\text{outer loop/learning meta model}}, \\ &\text{s.t. } \underbrace{\phi_{\tau,k+1} \leftarrow \phi_{\tau,k} - \alpha\nabla_{\phi_{\tau,k}}\ell_{\tau}^{\text{train}}}_{\text{inner loop/learning specific model}}, \underbrace{\phi_{\tau,0} = \Theta}_{\text{initialization}}, \end{aligned} \tag{1}$$

where a task-specific model ϕ_τ on task τ is initialized by the parameter of meta-model Θ and iteratively updated with K steps based on a supervision loss $\ell_{\tau}^{\text{train}}$ of a few

training samples in the inner loop. In the outer loop, the meta parameters Θ are updated by minimizing the test loss ℓ_τ^{test} over a batch of tasks $\mathbb{E}_{\tau\sim p(\tau)}$ with the gradient-adapted parameters $\phi_{\tau,k}$. MAML yields a meta-model that is explicitly fine-tuned to differentiate different instances.

Problem formulation. If all time series are recorded synchronously and the sensor number is fixed during the observation period, the data can be organized as a spatiotemporal matrix $\boldsymbol{X} \in \mathbb{R}^{N\times T}$ with $\boldsymbol{x}^i \in \mathbb{R}^T$ denoting the i-th series and $\boldsymbol{x}_t \in \mathbb{R}^N$ indicating observations at time t. Then the urban data imputation problem can be considered as estimating a probability $\forall \tau \in \{1,\dots,T\}, n \in \{i=1,\dots,N\}$:

$$p(\{\boldsymbol{x}_\tau^n | m_\tau^n = 0\} | \{\boldsymbol{x}_t^i | m_t^i = 1\}_{t=1,\dots,T}^{i=1,\dots,N}), \tag{2}$$

where $m_t^i \in \{0,1\}$ is an indicator of observable which is 1 if the measurements associated with the i-th sensor are valid at time step t. This posterior probability is estimated by a parameterized model (such as a neural network) by:

$$\Phi(\Omega(\boldsymbol{X})|\Theta) \approx \mathbb{E}[p(\{\boldsymbol{x}_\tau^n | m_\tau^n = 0\} | \{\boldsymbol{x}_t^i | m_t^i = 1\})], \tag{3}$$

where $\Omega(x_t^i) = x_t^i$ if $m_t^i = 1$. In the case of irregularly sampled data, each time series sampled from different cities or sensors can have different sampling frequencies and dimensions. We use a sequence of data pairs to denote this irregularity as $\boldsymbol{x}^i = \{(x_t^i, \xi^i, \tau_t)\}_{t=1}^{T_i}$ where the observation is paired with its location and time-index feature (ξ^i, τ_t). To mitigate the difficulty, we assume that the location of a sensor is fixed and the total number of sensors for a city remains unchanged during the period. Then the parameterized imputer is learned by minimizing the empirical loss over all time series in the training set:

$$\begin{aligned} \min_\Theta \mathcal{L}(\{\boldsymbol{x}^i\}_{i=1}^N; \Theta) &= \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[\ell(\boldsymbol{x}^i; \Theta)], \\ \text{with } \ell(\boldsymbol{x}^i; \Theta) &= \frac{1}{T_i}\sum_{t=1}^{T_i} \|x_t^i - \Phi_\Theta(\xi^i, \tau_t)\|_2^2, \end{aligned} \tag{4}$$

where $p_{\boldsymbol{x}}$ is the (unknown) data distribution. Φ is either a unified model or a composition of all individual models $\Phi = \{\Phi_1, \cdots, \Phi_N\}$, which corresponds to the **collaborative imputation** and **separate imputation** schemes respectively. To deal with irregular data, a continuous model Φ is expected to query data at arbitrary location and timestamp.

4 Methodology

This section elaborates the proposed model for cross-city collaborative imputation. We aim to learn an INR $\Phi_\Theta(\xi, \tau) : \mathbb{R}\times\mathbb{R} \mapsto \mathbb{R}$ that maps the timestamp (possibly with location) to the observed value x at each query coordinate. Then the continuous function $\Phi_\Theta(\cdot)$ allows for interpolation at arbitrary points, generating the imputation for irregular time series. The meta-learning method further enables it to generalize across heterogeneous cities. We term it Meta learning-based urban Time Series Imputer (MetaTSI). The overall architecture of MetaTSI is shown in Figure 1 (c).

4.1 Learning to Reconstruct Time Series Dynamics by Implicit Representation in the Embedding Space

4.1.1 Implicit Neural Representations for Time Series

Urban system involves complex spatiotemporal phenomena. The sensed time series may be generically generated from a high-dimensional dynamical system [28]:

$$\mathcal{C}(t, \boldsymbol{x}(t), \nabla_t \boldsymbol{x}(t), \nabla_t^2 \boldsymbol{x}(t), \dots) = 0, \tag{5}$$

where the time series $\boldsymbol{x}(t) : t \mapsto \mathbb{R}^m$ from a data generating process consists of successive, possibly irregular, observations of some dynamical process described by a system of partial differential equation $\mathcal{C}$. In this context, our goal is to learn a parameterized neural network Φ to map t to the target value $\boldsymbol{x}(t)$ while reconstructing the dynamics in Eq. 5, achieving the imputation in observations:

$$\min \mathcal{L} = \int_{\mathcal{T}} \|\mathcal{C}(\boldsymbol{x}(t), \Phi, \nabla_t \Phi, \nabla_t^2 \Phi, \dots)\| dt, \tag{6}$$

where $\mathcal{T}$ is the definition domain and the loss is feasible by sampling a dataset $\mathcal{D} = \{(\boldsymbol{x}(t), t) : t \in \mathcal{T}\}$ of coordinates and observations. Since Φ is implicitly defined by the relation modeled by $\mathcal{C}$, neural networks that parameterize such implicitly defined functions are referred to as INRs [21]. Typically, fitting an INR requires a large amount of observed data [21]. Directly learning INRs from sparse time series can be challenging and suboptimal due to the lack of physical guidance and data properties. To address this issue, we exploit methods that can guide the learning process.

4.1.2 Dynamics Reconstruction by the Embedding Theory

To obtain such a model Φ, we resort to the embedding theory [28], [29]. Generally, the dynamical system can be characterized by a state variable $\boldsymbol{s}(t) \in \mathbb{R}^n$ and a measurement function $\mathcal{H} : R^n \mapsto R^m$. Typically, $m < n$ as the state of the underlying dynamical process cannot be fully observed. For urban time series, we will consider in this paper the simplest case where $x(t)$ is a scalar (i.e. $m = 1$) time series. Given a partially observed system state $\boldsymbol{s}(t)$ with dynamics on state space $\mathcal{S} \subseteq \mathbb{R}^n$, it can be reconstructed by:

$$\vec{\boldsymbol{s}}(t) = f(\boldsymbol{s}(t)), \tag{7}$$

where $f(\cdot)$ is the reconstructor defined in the state space.

The measurement function $\mathcal{H}$ specifies the process of observing the system and extracting information evaluated at query time steps to generate a time-series:

$$x(t) = \mathcal{H}(\boldsymbol{s}(t)). \tag{8}$$

Then, an embedding is defined as a transformation that augments the observed time series by increasing its dimension with some time windows:

$$\Psi : \mathcal{R} \mapsto \mathcal{R}^{d_r},\ \vec{\boldsymbol{x}}(t) = \Psi(x(t)), \tag{9}$$

where $\vec{\boldsymbol{x}}(t)$ is the embedding vector with dynamics defined in a reconstructed state space $\vec{\mathcal{M}} \subseteq R^{d_r}$ and a transformed reconstructor $\mathcal{F}$. According to Takens' theory [29], if Ψ is a valid embedding, there exists a one-to-one mapping Ψ that we can identify a corresponding state point $x(t)$ on $\mathcal{S}$ for every $\vec{\boldsymbol{x}}(t)$ on $\vec{\mathcal{M}}$ via the inverse mapping $\Psi^{-1}(\vec{\boldsymbol{x}}(t))$:

$$\Psi^{-1} : \vec{\mathcal{M}} \mapsto \mathcal{S}, \Psi^{-1}(\vec{\boldsymbol{x}}(t)) = x(t), \tag{10}$$

and the the dynamics of the system are preserved by:

$$\mathcal{F} = \Psi \circ f \circ \Psi^{-1}. \tag{11}$$

We can find that learning the dynamics in the reconstructed state space $\vec{\mathcal{M}}$ is equivalent to learning the dy-

namics of the original system $\mathcal{S}$. Therefore, the time series imputation problem using embedding is posed as learning the parameterized reconstructor $\mathcal{F}_\Theta$ where:

$$\tilde{x}(t) = \Psi^{-1} \circ \mathcal{F}_\Theta \circ \Psi(x(t)). \tag{12}$$

As described in Eq. 12, the key to modeling the underlying dynamical system for imputation is to build a suitable data embedding Ψ and a mapping $\mathcal{F}$. Rather than relying on neural networks to directly process the raw data and correlate hidden states by temporal modules, we reconstruct the dynamical trajectory of the system using a nonparametric physical embedding as Ψ and parameterized INRs as $\mathcal{F}_\Theta$. We will elaborate on each of them in the following.

Remark (Relationship between data-driven time series modeling and implicit time series representation.)**.** *There are two principal ways to model time series. Data-driven time series models directly embed time series into hidden space using trainable transformation, relying on neural networks to model data correlations based on hidden representations. However, INRs embed the spatial-temporal coordinate in the state space using nonparametric techniques and reconstruct the dynamical structure, where observations are used as labels rather than input. According to Embedding Theory [29], [62], dynamical systems and time series can be mutually transformed through observation functions and reconstruction operation.*

4.1.3 Coordinate Delay Embedding

According to Takens' embedding theorem, it is guaranteed that a time delay embedding [63] with dynamics defined in a space of sufficiently large dimension $\mathbb{R}^{d_r}$ constructed from scalar time series is generically diffeomorphic to the full state space dynamics of the underlying system [64]. This facilitates the recovery of full-state dynamics using partially observed time series. To achieve this, time delay embedding augments a single scalar time series $x(t)$ to a higher dimension by constructing a delay vector $\vec{\boldsymbol{x}}(t) \in \mathbb{R}^m$:

$$\vec{\boldsymbol{x}}(t) = \Psi(x(t)) = [x(t), x(t+\delta), \ldots, x(t+(m-1)\delta)]^\mathsf{T}, \tag{13}$$

where δ is the delay lag, m is the embedding dimension, and $d_r = m$. Considering a given window T, the embedding matrix $\vec{\boldsymbol{X}} = \Psi(\boldsymbol{x}) \in \mathbb{R}^{(T-(m-1)\delta)\times m}$ is given by:

$$\vec{\boldsymbol{X}} = \Psi(\boldsymbol{x}) = [\Psi(x(1)), \Psi(x(2)), \ldots, \Psi(x(T-(m-1)\delta))]^\mathsf{T}, \tag{14}$$

which is expanded as:

$$\begin{bmatrix} x_1 & x_{1+\delta} & \cdots & x_{1+(m-1)\delta} \\ x_2 & x_{2+\delta} & \cdots & x_{2+(m-1)\delta} \\ x_3 & x_{3+\delta} & \cdots & x_{3+(m-1)\delta} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T-(m-1)\delta} & x_{T-(m-2)\delta} & \cdots & x_T \end{bmatrix}. \tag{15}$$

For a multivariate system with N components, one can reconstruct a topologically isomorphic manifold $\vec{\mathcal{M}}^i$ from every series $\boldsymbol{x}^i$ within the system $(i = 1, \ldots, N)$ in a d_r-dimensional space. This separate treatment aligns with the channel independence strategy for time series [11], [65].

Recall that the input to our model Φ is the coordinate in the reconstructed state space $\vec{\mathcal{M}}$. Different from the standard time delay embedding, we propose structuring the domain coordinate with delay embedding, which leads to the following relation in the state space:

$$\hat{\Psi}(\boldsymbol{x}) \approx \mathcal{F} \circ \Psi(\boldsymbol{\tau}) \in \mathbb{R}^{(T-(m-1)\delta)\times m}, \tag{16}$$

where $\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_T]^\mathsf{T} \in \mathcal{T}$ is the vector of time-index feature. Ψ organizes the index into a structured coordinate-delayed matrix, and $\mathcal{F}$ maps it to state values in higher dimensions. An inverse transform is adopted to embed the reconstruction to the original space through Eq. 12 as:

$$\Psi^{-1} \circ \hat{\Psi}(\boldsymbol{x}) = \boldsymbol{D}^\dagger \operatorname{vec}(\hat{\Psi}(\boldsymbol{x})) \in \mathbb{R}^T, \tag{17}$$

where $\boldsymbol{D} \in \{0,1\}^{(T-(m-1)\delta)\times m}$ is a duplication matrix and $\boldsymbol{D}^\dagger = (\boldsymbol{D}^\mathsf{T}\boldsymbol{D})^{-1}\boldsymbol{D}^\mathsf{T}$ is the pseudo-inverse of $\boldsymbol{D}$.

Our approach prescribes the global structure of the reconstructed space: $\vec{\mathcal{M}}^i$ also has an isomorphic topological structure with the original system and the reconstruction preserves the properties of the dynamical system that do not change under smooth coordinate changes, ensuring a continuous imputation function.

Remark (Relationship with deep time embedding)**.** *Many deep time embedding methods such as RNNs, casual attention, and reservoir computing, may also be related to delay embedding. In these models, the input time series is fed into a parameterized network that directly projects it into hidden states. The forward propagation of past states on current states effectively acts as a time delay embedding with small delay lag and large embedding dimension. Instead, our model reconstructs the dynamical trajectory of the system using physical priors as data embedding.*

4.1.4 Frequency-decomposed Multi-scale INRs

Next, we detail the architecture and parameterization of the reconstructor $\mathcal{F}_\Theta \doteq \Phi_\Theta$. It consists of two components: a mapping network and a modulation network. The former maps coordinates from the state space to target values in that location. The latter is used to adapt the model to different instances, which will be discussed in Section 4.2. Adhere to previous practices [25], we adopt a factorized functional representation to reduce the difficulty and complexity of learning in the entire space. By denoting the transformed coordinate in the embedding space as $\vec{\boldsymbol{\tau}}_t$, we have:

$$\begin{aligned} &\Phi_\Theta(\vec{\boldsymbol{\tau}}_t) = \mathcal{C} \times_1 \Phi_{\theta^1}(\vec{\tau}_t^1) \times_2 \Phi_{\theta^2}(\vec{\tau}_t^2), \forall (\vec{\tau}_t^1, \vec{\tau}_t^2) \in \vec{\boldsymbol{\tau}}_t, \\ &\Phi_{\theta^i} : \vec{\tau}_t^i \mapsto \Phi_{\theta^i}(\vec{\tau}_t^i) \in \mathbb{R}^{n_i}, \ \forall i \in \{1,2\}, \end{aligned} \tag{18}$$

where $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2}$ is the core tensor, $\times_i$ is the i-th mode tensor product, $\vec{\tau}_t^i$ is the embedded coordinate in the i-th axis, and $\Theta = \{\theta^1, \theta^2\} \cup \mathcal{C}$ are trainable parameters. Each Φ_{θ^i} can be defined separately to consider different patterns in the state space, and the interaction between the two components is preserved by $\mathcal{C}$. In practice, deep neural networks can become parameterization. However, learning to regress the target value in a high-dimensional system using the state coordinate is an ill-posed problem, as the associated neural tangent kernel is nonstationary [22], which is understood as the "spectral bias" issue.

There are several ways to alleviate this bias, including periodic activations [21], Fourier features [22], polynomial functions, and multiplicative filters [66]. We consider a multilayer and multiscale structure to exploit the complex spatiotemporal structure of urban data. Given $\forall \ell = \{1, \ldots, L\}$,

each of the mapping subnetwork $\Phi_\theta(\vec{\tau}_t^i) : \vec{\tau}_t^i \in \mathbb{R} \mapsto \Phi_\theta(\vec{\tau}_t^i) \in \mathbb{R}^{n_i}$ is formulated as:

$$\begin{aligned}
\boldsymbol{h}^{(1)} &= \texttt{ReLU}(\mathbf{W}^{(0)}\gamma_\sigma(\vec{\tau}_t^i) + \boldsymbol{b}^{(0)}),\\
\boldsymbol{h}^{(\ell+1)} &= \delta_i^{(\ell)} \odot \sin(\mathbf{W}^{(\ell)}\boldsymbol{h}^{(\ell)} + \boldsymbol{b}^{(\ell)} + \gamma_{\sigma_\ell}(\vec{\tau}_t^i)),\\
\tilde{\boldsymbol{h}}^{(L+1)} &= \mathbf{W}^{(L+1)}\boldsymbol{h}^{(L+1)} + \boldsymbol{b}^{(L+1)},
\end{aligned} \tag{19}$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_{(\ell+1)}\times d_{(\ell)}}, \mathbf{b}^{(\ell)} \in \mathbb{R}^{d_{(\ell+1)}}$ are layerwise parameters with $d_{(0)} = d_B N_f$ being the input dimension and $d_{(L+1)} = n_i$ being the output dimension, $\delta_i^{(\ell)} \in \mathbb{R}^{d_{(\ell+1)}}$ is the modulation variable in the ℓ-th layer described in Section 4.2.3, and $\odot$ is the element-wise product. $\gamma_\sigma(\cdot)$ is the concatenated Fourier features (CRF) with basis frequency $\mathbf{B}_k \in \mathbb{R}^{d_B/2\times 1}$ sampled from a Gaussian $\mathcal{N}(0, \sigma_k^2)$:

$$\begin{aligned}
\gamma(c)_\sigma = [&\sin(2\pi\mathbf{B}_1 c), \cos(2\pi\mathbf{B}_1 c), \dots,\\
&\sin(2\pi\mathbf{B}_{N_f} c), \cos(2\pi\mathbf{B}_{N_f} c)]^\mathsf{T} \in \mathbb{R}^{d_B N_f}.
\end{aligned} \tag{20}$$

There are several key points in Eq. 19 that need to be emphasized. (1) Both the sine activation and CRF are adopted to explicitly inject high-frequency structures into the network. (2) In particular, $\gamma_{\sigma_\ell}(\vec{\tau}_t^i)$ is the Fourier feature in the ℓ-th layer. Although CRF in the input can reduce the spectral bias, a simple stack of MLPs still suffers from capturing high-frequency data details [67], [68]. Instead, we decompose intermediate features into multiple frequency features to amplify high-frequency patterns by using different spectral bandwidths in different layers: $\sigma_1 \geq \sigma_2 \geq \dots \sigma_L$. Then the compositional nonlinearity of deep neural networks is applied recursively to progressively decode the multi-band intermediate features, achieving its representational complexity to model high-frequency patterns in deeper layers.

Consequently, a layerwise output is constructed that assembles the final reconstruction of Φ_{θ^i}:

$$\tilde{\boldsymbol{h}}^{(\ell)} = \texttt{ReLU}(\mathbf{W}_{\text{out}}^{(\ell)}\boldsymbol{h}^{(\ell)} + \boldsymbol{b}_{\text{out}}^{(\ell)}),\ \Phi_\theta(\vec{\tau}_t^i) = \sum_{\ell=1}^{L+1} \tilde{\boldsymbol{h}}^{(\ell)}. \tag{21}$$

Residual connections of all intermediate features into the output synthesize the multiscale reconstruction of the dynamical space and effectively predict details of data. Reconstruction from low-level to high-level frequency features resembles the coarse-to-fine approach in the spatial domain.

4.2 Cross-city Generalization by Meta Learning

We remark that an INR can encode a single time series by fitting to observed values. However, adopting individual INRs to memorize each instance in each city confronts two problems: (1) lack of sufficient observations in less developed cities poses challenges in model training; (2) having individual models for each location is computationally expensive and infeasible. A collaborative imputation scheme for learning across cities with limited resources that can be reused in new cities is preferable. However, learning across cities creates a consequent heterogeneity issue. Therefore, we resort to the meta learning scheme that allows us to learn the manifold in which the signals reside in a sensor-agnostic way, over arbitrary measurement from different locations. We further handle heterogeneity by proposing a normalization method and a modulation mechanism to condition the model to differentiate heterogeneous individual patterns.

4.2.1 Masked Instance Normalization

Urban time series feature large individual-level variations. Urban networks themselves are heterogeneous around the world [18]. Furthermore, mobility patterns within each city vary by space and time, causing the sensed urban time series to fluctuate drastically. To reduce individual-level variance for better generalization, we propose a masked instance normalization strategy. Given the observed time series $\boldsymbol{x}^i \in \mathbb{R}^{T_i}$, we normalize it before the delay embedding:

$$\begin{aligned}
\mathbb{E}_t[\boldsymbol{x}^i] &= \frac{1}{|\{m_t^i\}_{t=1}^{T_i}|}\sum_{t=1}^{T_i} m_t^i \odot x_t^i,\\
\text{Var}[\boldsymbol{x}^i] &= \frac{\sum_{t=1}^{T_i}(x_t^i - \mathbb{E}_t[\boldsymbol{x}^i])^2}{|\{m_t^i\}_{t=1}^{T_i}|}, \hat{x}_t^i = \frac{x_t^i - \mathbb{E}_t[\boldsymbol{x}^i]}{\sqrt{\text{Var}[\boldsymbol{x}^i] + \epsilon}},
\end{aligned} \tag{22}$$

where m_t^i is the masking indicator defined in Section 3, $\hat{x}_\tau^i$ is the normalized labels used for supervision. Normalized sequences can have a more consistent mean and variance, where the distribution discrepancy between different instances is reduced. This makes it easier for the model to learn local dynamics within the sequence while receiving input of consistent distributions. However, the input has statistics different from the original distribution. A denormalization step is needed to inform the model the original distribution of the instance by returning the distribution properties removed from the input to the model output:

$$\hat{\Phi}_\Theta(\tau_t) = \sqrt{\text{Var}[\boldsymbol{x}^i] + \epsilon} \cdot \Phi_\Theta(\tau_t) + \mathbb{E}_t[\boldsymbol{x}^i]. \tag{23}$$

4.2.2 Meta-learning-based Model Generalization

As stated above, a coordinate-based MLP (i.e., INR) can learn to represent each data instance, but the learned MLP cannot be generalized to represent other instances and requires re-optimizing from the scratch. Given a set of N data instances $\mathcal{X} = \{\boldsymbol{x}^i\}_{i=1}^N$ with $\boldsymbol{x}^i = \{(x_t^i, \tau_t)\}_{t=1}^{T_i}$, a simple approach to represent the entire dataset is to train an individual INR Φ_{θ^i} for each instance ($\mathcal{R}(\cdot)$ is omitted):

$$\ell_i(\boldsymbol{x}^i; \Theta^i) = \frac{1}{T_i}\sum_{t=1}^{T_i} \|x_t^i - \Phi_{\Theta^i}(\tau_t)\|_2^2, \tag{24}$$

where $\mathcal{H}_{\text{INR}} = \{\Phi(\tau_t; \Theta^i) | \Theta^i \in \Theta, i = 1, \dots, N\}$ is the hypothesis class of all INRs where Θ is the parameter space. However, such a hypothesis class is too expensive and computationally infeasible for a large number of instances in urban setting. In this case, standard INRs just memorize each data instance without generalization ability.

To enable INRs to be generalizable to different input instances in a memory-efficient way, we split the parameter space into two parts: (1) instance-specific parameters $\phi^i \in \Xi$, and (2) instance-agnostic parameters Θ. ϕ^i characterizes each data instance and aims to learn to adapt to specific patterns. Θ is shared across all instances and designed to learn the inductive bias, e.g., the underlying structural information in urban time series. Then, the loss of generalizable implicit neural representations (GINRs) is given as:

$$\begin{aligned}
&\min_{\Theta,\phi} \ell(\mathcal{X}; \Theta, \{\phi^i\}_{i=1}^N) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}[\ell^i(\boldsymbol{x}^i; \Theta, \phi^i)],\\
&= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{t=1}^{T_i} \|x_t^i - \Phi_{\Theta,\phi}(\tau_t; \phi^i)\|_2^2,
\end{aligned} \tag{25}$$

where $\phi_i \in \mathbb{R}^{d_{\text{latent}}}$ is the latent code for each data instance to account for the instance-specific data pattern.

Θ and ϕ^i characterize different perspectives of STTD and are nontrivial to obtain properly. To consider the per-instance variations, a natural way is to encode ϕ from the observed data and map it to parameters of a base network, e.g., all linear weights in Eq. 19. Consider the parameter that fully defines a L-layer single INR is $\boldsymbol{\theta} \in \mathbb{R}^D$ where $D = \sum_{\ell=1}^{L} d_{(\ell)}$, we can explicitly encode the observation $\boldsymbol{x}^i \in \mathbb{R}^{T_i}$ to a low-dimensional vector $\mathbf{c}^i \in \mathbb{R}^C$, then decode it to the hypothesis class of Φ with a hypernetwork h_{hyper}:

$$\begin{aligned} \texttt{Enc} &: \mathbb{R}^{T_i} \mapsto \mathbb{R}^C, \boldsymbol{x}^i \mapsto \texttt{Enc}(\boldsymbol{x}^i) := \mathbf{c}^i, \\ h_{\text{hyper}} &: \mathbb{R}^C \mapsto \mathbb{R}^D, \mathbf{c}^i \mapsto h_{\text{hyper}}(\mathbf{c}^i) = \boldsymbol{\theta}, \end{aligned} \tag{26}$$

where in this case $\phi = \{\mathbf{c}^i\}_{i=1}^N$ are the instance-specific parameters. However, manipulating the entire MLP can be parameter intensive and cause suboptimal representations. This process still admits a large parameter space as both the observation and hypernetwork can have a large dimension. In addition, since the input series can be sparse, heterogeneous, and length-varying, using a single hypernetwork as encoders to adapt the base network can be unpractical.

Instead, we treat ϕ^i as a learnable code and resort to the modulation method. We modulate the intermediate features of Φ_θ per instance using the compact form of ϕ^i. This is achieved by modifying only a few parameters of Φ_Θ through a modulation network discussed in Section 4.2.3. The conditioning modulations are processed as a function of ϕ^i, and each ϕ^i characterizes a specific instance. We can then *implicitly* obtain these latent codes as well as modulations using an *auto-decoding* mechanism, instead of the explicit encoding process in Eq. 26. We suppose that the internal manifold of correlated time series exists in a structured low-dimensional subspace that is globally consistent. Similar data samples should be embedded in a close location with small encoding steps. For data i, this is calculated by an iterative gradient descent process $\forall i = 1, \ldots, N$:

$$\phi^{(k+1),i} \leftarrow \phi^{(k),i} - \alpha \nabla_{\phi^i} \ell(\Phi_{\Theta, h_\omega(\phi)}, \{\boldsymbol{x}^i\}_{i \in \mathcal{B}}), \tag{27}$$

where α is the learning rate, h_ω is the hypernetwork that generates modulations from the latent code to condition INRs, and $\mathcal{B}$ is the sampled data batch. To initialize the learnable latent codes, a prior class over $p(\phi)$ is assumed to follow a zero-mean Gaussian. With this, the reduced hypothesis class is $\mathcal{H}_{\text{MetaTSI}} = \{\Phi(\tau; \Theta, \phi^i) | \Theta, \phi^i \in \mathbb{R}^{d_{\text{latent}}}\}$, which is more feasible in optimization. Generating the entire set of MLP parameters requires a large amount of memory. In contrast, the latent code is more parameter-efficient and accounts for only a small number of total parameters.

The intuition of Eq. 27 is to learn some priors over the function space of neural fields. However, learning all latent codes efficiently with a single base network is a challenge. Consequently, to integrate auto-decoding into the parameter learning procedure of the base network, a meta-learning scheme including inner loop and outer loop iterations is considered [30], as described in Eq. 1. The aim of meta-learning is to learn the base parameter conditioning on the latent code, so that the code can be auto-decoded in a small number of iterations. Therefore, (i) the inner loop learns to adapt ϕ to condition the base network Φ^i on the instance-specific pattern $\boldsymbol{x}^i$, and (ii) the outer loop learns to optimize the shared base parameters. In practice, the latent code is optimized by a small number of gradient descent steps in the inner loop (e.g., $N_{\text{inner}} = 3$ is sufficient). Additionally, multivariate correlations are implicitly modeled by the interaction of latent code during optimization.

4.2.3 Hierarchical Modulation Mechanism

We now introduce the modulation network. Recall that the key of MAML is the fast adaptation to local variations with minimal modification of network weights. However, due to high heterogeneity of spatiotemporal patterns of urban data, manipulating only a single weight matrix is inadequate for complex datasets. To this end, we propose to adapt the model to different instances by modulating hierarchical hidden activations of the mapping subnetwork in Eq. 19. Motivated by [47], we model ϕ as a series of latent codes $\{\phi^i \in \mathbb{R}^{d_{\text{latent}}}\}_{i=1}^N$ for each instance to account for the instance-specific data pattern and make Φ_θ a base network conditional on the latent code ϕ. The per-sample modulations $\boldsymbol{s}^i$ are considered as a function conditioned on the latent code ϕ^i that represents the individual time series:

$$\begin{aligned} \delta^{(\ell),i} &= \boldsymbol{s}^{(\ell),i} + \delta^{(\ell-1),i}, \ \ell = \{1, \ldots, L\}, \\ \boldsymbol{s}^{(\ell),i} &= h_\omega^{(\ell)}(\phi^i) = \sigma(\mathbf{W}_s^{(\ell)} \phi^i + \boldsymbol{b}_s^{(\ell)}), \\ \delta^{(0),i} &= \mathbf{W}_s^{(0)} \phi^i + \boldsymbol{b}_s^{(0)}, \end{aligned} \tag{28}$$

where $\delta^{(\ell),i}$ is the modulation vector added to the pre-activation at layer ℓ of instance i, which enables one to modulate the weight space in a low-dimensional space, thereby significantly reducing the parameters. Crucially, we allocate more expressive hypernetwork modules to high-frequency layers: for $\ell \geq \ell_{\text{HF}}$, $h_\omega^{(\ell)}$ is implemented as a one-layer MLP with ReLU nonlinearity, whereas for low-frequency layers ($\ell < \ell_{\text{HF}}$) it remains a single linear projection, i.e., $\sigma = \mathbb{I}$. This design ensures that deeper (more high-frequency) feature maps which capture rapid spatiotemporal fluctuations are modulated by richer nonlinear transforms, while coarse (low-frequency) features receive lighter and more stable shifts. Here, $h_\omega^{(\ell)}(\cdot | \omega \in \theta) : \mathbb{R}^{d_{\text{latent}}} \mapsto \mathbb{R}^{d_{(\ell)}}$ shares parameters across instances but varies per layer, enabling a layer-wise hierarchy of modulation bandwidths. Low-frequency modulations (early layers) shift broad trends with minimal distortion, whereas high-frequency modulations (later layers) can create fine-grained, localized patterns through extra nonlinear capacity. Note that the parameters $\{\mathbf{W}_s^{(\ell)}, \mathbf{b}_s^{(\ell)}\}_{\ell=1}^L$ of h_{hyper} is a subset of θ.

As can be seen, the modulation is established in a skip-connected structure: each layer's correction $\delta^{(\ell),i}$ accumulates previous shifts $\delta^{(\ell-1),i}$, yielding a smooth interpolation between coarse (trend) and fine (fluctuation) adjustments. In addition, the differential hypernetwork depth directly controls the receptive bandwidth of modulation signals: lower layers enforce global consistency, higher layers encode local heterogeneities, which hierarchically modulate complex signals. Finally, per layer modulation is injected into hidden states of the base network via Eq. 19:

$$\boldsymbol{h}^{(\ell+1)} = \delta_i^{(\ell)} \odot \sin(\mathbf{W}^{(\ell)} \boldsymbol{h}^{(\ell)} + \boldsymbol{b}^{(\ell)} + \gamma_{\sigma_\ell}(\vec{\tau}_t^i)).$$

In summary, we provide the workflow of MetaTSI that includes both the training stage and the inference stage in Algorithm 1. During training, the knowledge from source cities is encoded in weights of deep neural networks and inform the fast adaptation of downstream cities. During inference, the latent code ϕ adapts quickly to the current data pattern in just a few gradient steps. This mechanism makes MetaTSI capable of generalizing to new cities efficiently, which is a significant advantage over individual models.

Algorithm 1: MetaTSI for Cross-city Learning

Input: Base network parameter Θ, hypernetwork parameter ω, training dataset $\mathcal{X}$.
Output: Trained base model Φ_Θ and latent codes for all instances $\{\phi^i\}_{i=1}^N$.

// Model training stage
while *not convergence* **do**
  Sample a batch $\mathcal{B}$ of data $\{\{(x_t^i, \tau_t)\}_{t=1}^{T_i}\}_{i\in\mathcal{B}}$;
  Normalize the observation $\hat{x}_t^i$ using Eq. 22;
  Embed $\{\{(\hat{x}_t^i, \tau_t)\}_{t=1}^{T_i}\}_{i\in\mathcal{B}}$ using Eq. 14;
  Set latent code to zeros $\phi^i \leftarrow 0, \forall i \in \mathcal{B}$;
  // Inner loop for latent modulations
  for $s = 1 : N_{inner}$ *and* $i \in \mathcal{B}$ **do**
    $\phi^i \leftarrow \phi^i - \alpha\nabla_\phi \ell(\Phi_{\Theta,h_\omega(\phi)}, \Psi(\{(\hat{x}_t^i, \tau_t)\}_{t=1}^{T_i}))|_{\phi=\phi^i}$;
  // Outer loop for updating base parameters
  $[\Theta, \omega] \leftarrow [\Theta, \omega] - \eta\nabla_{\Theta,\omega} \frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}} \ell(\Phi_{\Theta,h_\omega(\phi)}, \Psi(\{(\hat{x}_t^i, \tau_t)\}_{t=1}^{T_i}))|_{\phi=\phi^j}$;
  Recover outputs to the original space using Eq. 17;
  De-normalize the reconstruction using Eq. 23;
// Model inference stage
Given a partially observed new instance in grid $\mathcal{M}$ $\{\tau_t^\star, \boldsymbol{x}_t^\star\}_{t\in\mathcal{M}}$, set $\phi^\star \leftarrow 0$;
Perform normalization and embedding;
for $s = 1 : N_{inner}$ **do**
  $\phi^\star \leftarrow \phi^\star - \alpha\nabla_\phi \ell(\Phi_{\theta,h_\omega(\phi^\star)}, \{\tau_t^\star, \boldsymbol{x}_t^\star\}_{t\in\mathcal{M}}|_{\phi=\phi^\star}$;
Evaluate $\Phi_{\theta,h_\omega(\phi^\star)}(\tau^\star)$ for any $\tau^\star \in \mathcal{M}$.

4.3 Algorithmic Analysis

This section provides further discussion on the architectural bias of MetaTSI. We theoretically demonstrate that the mapping network is a continuous function of the target signal and can be viewed as a composition of Fourier series.

4.3.1 MetaTSI as Continuous Functions

To examine whether MetaTSI can encode continuous functions and enable coordinate interpolation, we evaluate its continuity. We start by supposing that the Lipschitz constant of sine activation is ϵ, all bias terms are absorbed in weight matrices, and the ℓ_1 norm of weight matrices, the core tensor $\mathcal{C}$, as well as all latent vectors ϕ are bounded by ξ, λ, and η respectively. Then $\forall\Phi(\tau)$ we have:

$$\begin{aligned}|\Phi(\tau)| \le |\mathbf{W}^{(L+1)}||(\mathbf{W}_s^L\phi) \odot \sin(\mathbf{W}^L \cdots \\ \cdots (\mathbf{W}_s^1\phi) \odot \sin(\mathbf{W}^1))||\tau| \le \epsilon^L\xi^{2L+1}\eta^L|\tau|.\end{aligned} \tag{29}$$

Then for the factorized MetaTSI model $\Phi_\Theta(\vec{\boldsymbol{\tau}}) = \mathcal{C} \times_1 \Phi_{\theta_1}(\vec{\tau}^1) \times_2 \Phi_{\theta_2}(\vec{\tau}^2)$, we have:

$$\begin{aligned}|\Phi(\vec{\tau}^1, \vec{\tau}^2) - \Phi(\vec{\tau}^{1'}, \vec{\tau}^2)| \le \lambda\epsilon^{2L}\xi^{4L+2}\eta^{2L}\nu|\vec{\tau}^1 - \vec{\tau}^{1'}|, \\ |\Phi(\vec{\tau}^1, \vec{\tau}^2) - \Phi(\vec{\tau}^1, \vec{\tau}^{2'})| \le \lambda\epsilon^{2L}\xi^{4L+2}\eta^{2L}\nu|\vec{\tau}^2 - \vec{\tau}^{2'}|,\end{aligned} \tag{30}$$

where $\nu = \max\{\vec{\tau}^1, \vec{\tau}^2\}$ and the above inequality holds based on the fact that $|a \odot b| \le |a||b|$. The above discussion indicates that Φ is Lipschitz continuous in the input coordinate system, serving as a smooth function approximator.

4.3.2 MetaTSI as Fourier Series

To answer the research question of why MetaTSI is effective for time series imputation, we explore its property through Fourier analysis. We show that hidden representations in the modulated mapping network are equivalent to a composition of Fourier series, which naturally characterizes the representation of the underlying signal.

First, we indicate that the CRF in Eq. 20 is essentially a sine term. Considering the simplest case that $N_f = 1$, i.e.,

$$\gamma(\tau) = [\sin(2\pi\mathbf{B}\tau), \cos(2\pi\mathbf{B}\tau)]^\mathsf{T}. \tag{31}$$

Then it passes through the first layer of Eq. 19:

$$\begin{aligned}\boldsymbol{h}^{(1)} &= \mathbf{W}^{(0)}\gamma_\sigma(\tau) + \boldsymbol{b}^{(0)}, \\ &= \mathbf{W}^{(0)}[\sin(2\pi\mathbf{B}\tau), \cos(2\pi\mathbf{B}\tau)]^\mathsf{T} + \boldsymbol{b}^{(0)}, \\ &= \mathbf{W}^{(0)}\sin(2\pi\mathbf{B}'\tau + \rho) + \boldsymbol{b}^{(0)},\end{aligned} \tag{32}$$

where $\mathbf{B}' = [\mathbf{B}, \mathbf{B}]^\mathsf{T}$ and $\rho = [\pi/2, \ldots, \pi/2, 0, \ldots, 0]$. This shows that it is equivalent to a single-layer network with sine activation. The case $N_f \neq 1$ can be verified similarly.

Second, the output of subsequent layers can be approximated as a combination of sinusoidal bases:

$$\boldsymbol{h}^{(\ell+1)} \approx \sum_{k=1}^{K} \bar{\alpha}_k \sin(2\pi\bar{\omega}_k\boldsymbol{h}^{(\ell)} + \bar{\rho}_k) + \bar{\beta}_k, \tag{33}$$

where K is the total order of coefficients $\bar{\alpha}_k$, frequencies $2\pi\bar{\omega}_k$, phase shifts $\bar{\rho}_k$, and bias $\bar{\beta}_k$. This is evidenced by:

$$\begin{aligned}\boldsymbol{h}^{(\ell+1)} &= \delta^{(\ell),i} \odot \sin(\mathbf{W}^{(\ell)}\boldsymbol{h}^{(\ell)} + \boldsymbol{b}^{(\ell)}), \\ &= \delta^{(\ell),i} \odot \sin(\mathbf{W}^{(\ell)}(\delta^{(\ell-1),i} \odot \sin(\ldots)) + \boldsymbol{b}^{(\ell)}), \\ &= (\widetilde{\mathbf{W}}^{(\ell)} \circ \sin \circ \cdots \circ \widetilde{\mathbf{W}}^{(0)})(\tau),\end{aligned} \tag{34}$$

where the bias term and element-wise product are absorbed in a new weight matrix $\widetilde{\mathbf{W}}^{(\ell)}$. As sinusoidal activation can be effectively approximated using polynomials with a Taylor expansion, the composition of sinusoidal signals is still a set of wave signals. This makes the hidden state resemble a Fourier representation of the underlying signal.

Last, by observing Eq. 34, we find that the modulation vector $\delta^{(\ell),i}$ affects the amplitude, phase shift, and frequency in each hidden layer of the mapping network. This property guarantees the expressivity of the latent modulation, making MetaTSI generalizable to large-scale complex datasets.

4.3.3 Theoretical Analysis of Spectral Bias Reduction

While our frequency-decomposed multi-scale MLP (Eq. 19) empirically captures both coarse and fine temporal patterns, we now theoretically show how the interplay of Fourier features and sine activations explicitly mitigates the well-known spectral bias of standard MLPs [21], [22].

Spectral bias in standard MLPs. Following the neural tangent kernel (NTK) framework [69], the training dynamics of an L-layer MLP $f(\cdot;\theta)$ under squared error can be linearized around initialization, yielding:

$$\hat{\mathbf{y}}^{(t)} \approx \mathbf{K}_{\text{test}}\mathbf{K}^{-1}(\mathbf{I} - e^{-\eta\mathbf{K}t})\mathbf{y}, \tag{35}$$

where $\mathbf{K}_{ij} = k_{\text{NTK}}(\mathbf{x}_i, \mathbf{x}_j)$ is the NTK matrix on training inputs and η is the learning rate [69]. Its eigendecomposition $\mathbf{K} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ reveals that models with larger eigenvalues λ converge faster:

$$\left|\mathbf{Q}^\top(\hat{\mathbf{y}}^{(t)}_{\text{train}} - \mathbf{y})\right| \approx \left|e^{-\eta\,\Lambda\,t}\,\mathbf{Q}^\top\mathbf{y}\right|. \tag{36}$$

Since the NTKs of MLPs decay rapidly in the frequency domain, high-frequency components (small λ) are learned only slowly. This is the *spectral bias* problem.

Fourier features induce a wider NTK spectrum. Injecting Fourier features via $\gamma_\sigma(\tau)$ (Eq. 20) converts the input kernel from a dot-product one into a stationary, shift-invariant one:

$$k_\gamma(\tau_i, \tau_j) = \gamma_\sigma(\tau_i)^\top\gamma_\sigma(\tau_j) = \sum_{j=1}^{N_f}\cos\big(2\pi B_j(\tau_i - \tau_j)\big)\,. \tag{37}$$

As shown in [22], the composed NTK satisfies:

$$\begin{aligned} k_{\text{NTK}}\big(\gamma_\sigma(\tau_i), \gamma_\sigma(\tau_j)\big) =& g_{\text{NTK}}\big(\gamma_\sigma(\tau_i)^\top\gamma_\sigma(\tau_j)\big) \\ =& g_{\text{NTK}}\big(g_\gamma(\tau_i - \tau_j)\big), \end{aligned} \tag{38}$$

yielding a kernel whose spectral density decays far more slowly than that of the vanilla NTK of MLPs. Thus, high-frequency components are assigned larger eigenvalues and are learned more rapidly.

Sine activations as implicit Fourier mappings. Moreover, the use of sine activations in each hidden layer (Eq. 19) is formally equivalent to learning with an explicit Fourier basis. For example, a two-layer subnetwork

$$F_s(v) = W\,\sin\big(2\pi\,W_s\,v + b_s\big) + b$$

can be reparametrized (by choosing b_s appropriately) into

$$\begin{aligned} F_f(v) =& W_f\big[\cos(2\pi\,W_{f_1}v),\,\sin(2\pi\,W_{f_1}v),\ldots, \\ & \cos(2\pi\,W_{f_N}v),\,\sin(2\pi\,W_{f_N}v)\big] + b, \end{aligned} \tag{39}$$

i.e. a linear readout over Fourier features [21]. Hence each sine layer further enriches the network's effective NTK spectrum with additional high-frequency components.

Multi-scale composition for progressive decoding. By stacking CRF $\gamma_{\sigma_\ell}(\tau)$ with non-increasing bandwidths $\sigma_1 \geq \cdots \geq \sigma_L$ and interleaving sine layers (Eq. 19), we ensure:

$$\text{NTK}_{\text{total}} = h_{\text{NTK}} \circ h_{\gamma_{\sigma_L}} \circ \cdots \circ h_{\gamma_{\sigma_1}}, \tag{40}$$

which exhibits an ultra-wide spectral support across scales. Early layers capture coarse (low/σ_1) frequencies, while deeper layers recover fine (high/σ_L) detail, yielding a coarse-to-fine reconstruction that converges rapidly at all frequencies. Together, the CRF input mapping and periodic activations guarantee that our mapping network's NTK retains a broad, slowly decaying spectrum, directly alleviating the spectral bias of vanilla MLPs. The multi-scale design then composes these effects to progressively decode high-frequency dynamics, providing both strong expressiveness and fast convergence on complex temporal signals.

5 Experiments

This section designs a battery of experiments to evaluate whether MetaTSI can provide high-performance, generalizable, and computationally efficient imputation for urban time series. We (1) compare it with state-of-the-art imputation models on single-city benchmarks; (2) demonstrate generalization across cites and sensors; (3) assess its few-shot performance in unseen samples; and (4) detail additional properties of MetaTSI, such as computational efficiency, architectural rationality, and interpretable latent manifolds.

TABLE 1: Statistics of cross-city datasets.

City	# of Sensors	Seq. Len.	Frequency
London	797	1440	5 min
Orange	344	1344	15 min
Utrecht	407	1152	5 min
Los Angeles	702	1344	3 min
San Diego	702	1344	15 min
Riverside	474	1344	15 min
San Francisco	259	1344	15 min
Melbourne	926	672	15 min
Bern	462	2015	5 min
Kassel	292	1171	5 min
Darmstadt	112	1936	3 min
Toronto	184	672	15 min
Speyer	63	2016	3 min
Manchester	148	1911	5 min
Luzern	101	3360	3 min
Taipei	379	3280	3 min
Contra Costa	448	1344	15 min
Hamburg	254	2016	3 min
Munich	437	288	5 min
Zurich	803	2016	3 min

5.1 Datasets and Experimental Settings

Datasets. To scale the analysis and evaluation to cover multiple cities across the world, we collect and construct a large-scale urban traffic benchmark. This benchmark includes 20 cites worldwide, with traffic flow data processed from UTD19 data [1] and PeMS data [2]. UTD19 consists mainly of measurements from loop detectors from 2017-2019, which record vehicle flow and occupancy in a relatively small aggregation interval, typically 3-5min. The cities included in UTD19 are located mainly in European countries. PeMS data is collected from individual detectors spanning the freeway system across all major metropolitan areas of California, which is processed into a regular interval of 5 minutes. The whole dataset includes more than 8,000 heterogeneous time series with different lengths and frequencies. Brief summary of the data is given in Table. 1. To compare with existing models, we ensure that time series from the same city have the same length. For a straightforward evaluation, we report the MAE and MSE metrics on Z-normalized values as different series have different scales and dimensions.

Baselines. We compare MetaTSI with both analytical models and deep implicit representation models. They are from state-of-the-art venues in related literature. For single-city comparison, we consider: (1) SIREN (NeurIPS'20) [21]; (2) FourierNet (NeurIPS'20) [22]; (3) DeepTime (ICML'23) [26]; (4) TimeFlow (TMLR'24) [42]; (5) LCR (TKDE'24) [13]; (6) TRMF (NeurIPS'16) [12]; (7) TIDER (ICLR'23) [70]; (8) mTAN (ICLR'21) [71]; (9) LRTFR (TPAMI'23) [25]. For cross-city experiments, we consider generalizable models: (1) TimeFlow; (2) Functa (ICML'22) [47]; (3) DeepTime; (4) HyperNet-SIREN; (5) SIREN+. Note that since the data has large spatial dimensions and irregular sampling frequencies, many deep imputation models (e.g., BRITS, SAITS, and ImputeFormer) are not well suited for this challenging task

1. https://utd19.ethz.ch/
2. https://pems.dot.ca.gov/

TABLE 2: Result (normalized metrics) in London, Utrecht, Manchester, San Francisco, Melbourne and Toronto.

		London		Utrecht		Manchester		San Francisco		Melbourne		Toronto	
	Models	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
20% observation rate	Average	0.485	0.438	0.685	0.874	0.533	0.531	0.816	0.998	0.649	0.766	0.562	0.582
	TRMF (NeurIPS'16)	0.245	0.129	0.324	0.236	0.139	0.056	0.157	0.099	0.245	0.196	0.199	0.095
	TIDER (ICLR'23)	0.282	0.179	0.363	0.275	0.206	0.105	0.273	0.150	0.245	0.124	0.322	0.212
	mTAN (ICLR'21)	0.345	0.188	0.491	0.552	0.387	0.303	0.567	0.490	0.498	0.527	0.679	0.888
	SIREN (NeurIPS'20)	0.674	0.847	0.529	0.629	0.668	0.773	0.341	0.235	0.515	0.531	0.749	0.984
	FourierNet (NeurIPS'20)	0.627	0.701	0.334	0.248	0.155	0.061	0.162	0.062	0.530	0.548	0.358	0.316
	DeepTime (ICML'23)	0.491	0.526	0.474	0.452	0.451	0.425	0.526	0.527	0.535	0.611	0.464	0.508
	TimeFlow (TMLR'24)	0.287	0.181	0.347	0.262	0.255	0.157	0.347	0.224	0.311	0.248	0.393	0.322
	LRTFR (TPAMI'23)	0.256	0.196	0.310	0.218	0.335	0.233	0.327	0.217	0.229	0.111	0.295	0.193
	LCR-1D (TKDE'24)	0.267	0.134	0.320	0.208	0.161	0.064	0.155	0.051	0.228	0.103	0.195	0.085
	LCR-2D (TKDE'24)	0.291	0.151	0.329	0.211	0.189	0.077	0.143	0.042	0.254	0.121	0.212	0.092
	MetaTSI-1D (ours)	0.234	0.121	0.266	0.154	0.159	0.061	0.181	0.077	0.244	0.122	0.202	0.098
	MetaTSI-2D (ours)	**0.212**	**0.091**	**0.257**	**0.148**	**0.115**	**0.045**	**0.131**	**0.040**	**0.183**	**0.071**	**0.159**	**0.065**
10% observation rate	Average	0.486	0.439	0.684	0.879	0.533	0.534	0.820	1.005	0.651	0.772	0.563	0.593
	TRMF (NeurIPS'16)	0.279	0.161	0.352	0.259	0.188	0.120	0.189	0.096	0.278	0.160	0.282	0.174
	TIDER (ICLR'23)	0.384	0.299	0.467	0.420	0.324	0.224	0.430	0.343	0.423	0.335	0.487	0.467
	mTAN (ICLR'21)	0.352	0.194	0.390	0.292	0.411	0.389	0.593	0.550	0.564	0.589	0.701	0.958
	SIREN (NeurIPS'20)	0.692	0.900	0.534	0.640	0.701	0.892	0.385	0.310	0.527	0.609	0.723	1.063
	FourierNet (NeurIPS'20)	0.648	0.784	0.382	0.342	0.261	0.171	0.237	0.135	0.535	0.556	0.456	0.544
	DeepTime (ICML'23)	0.567	0.662	0.610	0.720	0.635	0.763	0.740	0.930	0.628	0.790	0.542	0.652
	TimeFlow (TMLR'24)	0.316	0.208	0.362	0.273	0.288	0.187	0.430	0.342	0.340	0.262	0.336	0.262
	LRTFR (TPAMI'23)	0.270	0.197	0.323	0.262	0.343	0.251	0.354	0.246	0.237	0.122	0.310	0.213
	LCR-1D (TKDE'24)	0.293	0.160	0.342	0.226	0.199	0.088	0.204	0.083	0.282	0.151	0.265	0.142
	LCR-2D (TKDE'24)	0.303	0.165	0.339	0.217	0.212	0.092	0.191	0.072	0.288	0.152	0.260	0.131
	MetaTSI-1D (ours)	0.238	0.125	0.281	0.169	0.146	0.066	0.239	0.125	0.291	0.186	0.275	0.191
	MetaTSI-2D (ours)	**0.220**	**0.102**	**0.272**	**0.192**	**0.123**	**0.065**	**0.151**	**0.058**	**0.198**	**0.088**	**0.185**	**0.088**
5% observation rate	Average	0.488	0.443	0.688	0.888	0.531	0.537	0.824	1.017	0.657	0.786	0.564	0.603
	TRMF (NeurIPS'16)	0.320	0.219	0.380	0.298	0.275	0.182	0.296	0.188	0.387	0.272	0.333	0.225
	TIDER (ICLR'23)	0.454	0.389	0.562	0.608	0.533	0.579	0.674	0.785	0.629	0.713	0.719	1.032
	mTAN (ICLR'21)	0.441	0.395	0.580	0.662	0.683	0.810	0.789	1.008	0.580	0.626	0.754	1.113
	SIREN (NeurIPS'20)	0.721	1.220	0.558	0.725	0.743	1.023	0.407	0.357	0.591	0.757	0.749	1.140
	FourierNet (NeurIPS'20)	0.690	0.937	0.462	0.530	0.267	0.210	0.358	0.286	0.601	0.783	0.611	0.843
	DeepTime (ICML'23)	0.654	0.831	0.752	1.038	0.739	0.995	0.861	1.241	0.713	0.984	0.782	1.234
	TimeFlow (TMLR'24)	0.338	0.238	0.586	0.773	0.321	0.224	0.611	0.655	0.373	0.307	0.407	0.362
	LRTFR (TPAMI'23)	0.337	0.293	0.376	0.361	0.358	0.281	0.431	0.359	0.279	0.174	0.345	0.267
	LCR-1D (TKDE'24)	0.337	0.210	0.380	0.268	0.251	0.127	0.266	0.136	0.354	0.230	0.364	0.249
	LCR-2D (TKDE'24)	0.335	0.204	0.371	0.252	0.252	0.123	0.242	0.113	0.342	0.209	0.324	0.193
	MetaTSI-1D (ours)	0.262	0.148	0.337	0.245	0.177	0.098	0.353	0.266	0.395	0.372	0.397	0.359
	MetaTSI-2D (ours)	**0.244**	**0.144**	**0.306**	**0.298**	**0.139**	**0.095**	**0.182**	**0.083**	**0.243**	**0.146**	**0.227**	**0.136**

and we do not include them as baselines. In addition, both LCR and MetaTSI have two variations according to model implementations: (1) 1D model: treating each time series as individuals and modeling independently; (2) 2D model: treating the multivariate time series as a whole matrix and adding an additional spatial dimension. Fig. 2 provides a diagram to compare the difference of the 1D and 2D models.

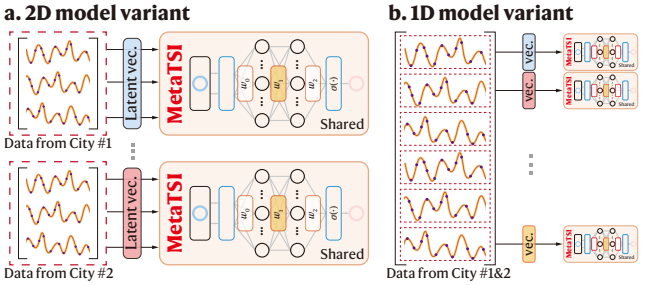


Fig. 2: Illustration of the 1D and 2D model variants. (a) 2D model treats all time series from one city as a whole matrix and adapts to different cities with city-specific embeddings. (b) 1D model is optimized over the set of all series from all cities, and adapts to each series by series-wise embedding.

Hyperparameters. There are several key hyperparameters in MetaTSI. For CRF, the number of Fourier features N_f is set to 2048, the initial scales are set to $[0.01, 0.1, 1, 5, 10, 20, 50, 100]$, and the layerwise frequency scales are set to $[10, 5, 1, 0.1, 0.01]$. For the mapping and modulation network, the layer number is set to 5 with a hidden dimension of 1024 and the delay embedding dimension is chosen from $[24, 36, 72]$. For optimization, the Adam optimizer is adopted with an outer learning rate of 5×10^{-4} and an inner learning rate of 1×10^{-2}. The dimension of the latent code is set to 128 and the number of inner steps is $3 \sim 5$. For baseline models, as few of them were originally designed for irregular time series imputation, we determine their respective hyperparameters using cross-validation.

5.2 Model Comparison within a Single City

In the first experiment, we train each model in each city with varying observation rates (i.e., $5\%, 10\%, 20\%$). According to their mechanisms, models either are fitted to the observations and predict missing values according to the coordinates, or are optimized to reconstruct the distributions or patterns of the observations. Results in Table. 2 indicate that the proposed MetaTSI-2D model consistently achieves better performances than baselines in all cities, and MetaTSI-1D also achieves performance comparable to the state-of-the-art. When there are fewer observations, the superiority of MetaTSI is more significant. In particular, MetaTSI-2D performs better than MetaTSI-1D in fitting observations within a single city. This is achieved by the strong spatial inductive bias imposed on the model, which leads to a larger model capacity to fit data distributions. However, in the next section we show that this sacrifices model generalizability.

5.3 Generalization across Multiple Cities

Second, we examine the generalizability of models. Our objective is to learn a unified model to reconstruct all

TABLE 3: Cross-city learning results of different models (normalized MAE and MSE).

Models	**MetaTSI-1D**		**MetaTSI-2D**		TimeFlow		SIREN+		Functa		ST-GFSL		TBP	
Cities	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
London	**0.253**	**0.132**	0.302	0.171	0.516	0.499	0.488	0.443	0.514	0.500	0.803	1.058	0.897	1.338
Orange	**0.238**	**0.108**	0.240	0.103	0.514	0.424	0.864	1.125	0.373	0.236	0.428	0.312	0.532	0.454
Utrecht	**0.278**	**0.164**	0.332	0.244	0.730	1.026	0.667	0.836	0.734	1.057	0.656	0.773	0.621	0.702
Los Angeles	**0.184**	**0.074**	0.233	0.107	0.518	0.439	0.804	0.977	0.410	0.282	0.548	0.504	0.621	0.606
San Diego	**0.174**	**0.066**	0.265	0.135	0.508	0.423	0.795	0.952	0.421	0.295	0.539	0.499	0.603	0.557
Riverside	**0.182**	**0.073**	0.209	0.091	0.447	0.332	0.714	0.774	0.351	0.209	0.544	0.509	0.672	0.758
San Francisco	**0.242**	**0.114**	0.294	0.164	0.548	0.480	0.811	0.991	0.456	0.349	0.557	0.524	0.642	0.659
Melbourne	**0.329**	**0.243**	0.360	0.247	0.484	0.449	0.640	0.748	0.411	0.337	0.748	0.898	0.781	0.973
Bern	**0.291**	**0.178**	0.327	0.220	0.781	1.096	0.801	1.079	0.730	1.014	0.623	0.698	0.563	0.556
Kassel	**0.329**	**0.268**	0.403	0.363	0.728	0.979	0.705	0.881	0.720	0.976	0.618	0.700	0.702	0.870
Darmstadt	**0.389**	**0.327**	0.418	0.348	0.712	0.883	0.733	0.888	0.696	0.863	0.835	1.116	0.837	1.168
Toronto	**0.367**	**0.298**	0.322	0.211	0.506	0.474	0.604	0.647	0.479	0.437	0.884	1.243	0.903	1.228
Speyer	**0.455**	**0.387**	0.443	0.344	0.873	1.306	0.943	1.410	0.834	1.229	0.588	0.575	0.557	0.505
Manchester	**0.173**	**0.067**	0.196	0.094	0.607	0.698	0.515	0.487	0.666	0.813	0.887	1.244	1.010	1.721
Luzern	**0.341**	**0.203**	0.340	0.204	0.940	1.400	0.933	1.288	0.962	1.482	0.656	0.773	0.723	0.868
Average	**0.282**	**0.181**	0.313	0.204	0.627	0.727	0.734	0.902	0.584	0.672	0.682	0.812	0.711	0.864

time series simultaneously for all cities. We select 15 cities for this experiment and the data missing rate for each city is randomly sampled from $[80\%, 95\%]$. As data are irregularly sampled in different cities, we need continuous models to deal with the irregularity. Therefore, only continuous models equipped with transfer-learning-related strategies can complete this task. We also compare our models with non-INR architectures discussed in Section 2. However, most of them are not directly applicable to our experiments: (1) discrete methods such as MetaST [57], CrossTReS [59], and Mest-GAN [60] are grid-based and cannot be applied to varying spatiotemporal dimensions; (2) ST-GFSL [58] and TPB [61] are designed for time series forecasting with complete historical-future data pairs. Therefore, we modify ST-GFSL and TPB to adapt imputation tasks. Results of different methods are given in Table. 3. It is observed that in this case MetaTSI-2D is surpassed by MetaTSI-1D which has better generalizability. Different cities have heterogeneous spatial patterns such as different network topologies and mobility demands. Learning a city as a whole increases the challenges of model-based transfer. Other meta-learning-based INR models cannot achieve competitive performances, which demonstrate the effectiveness of our multi-scale architecture and modulation strategy particularly designed for urban time series. Non-INR methods (ST-GFSL and TBP) show inferior performances as they have no suitable mechanism for reconstructing sparse data.

5.4 Generalization on Unseen Instances

Next, we evaluate the out-of-distribution generalization performance of models. This scenario happens often in practice when the pretrained model is applied to new locations or cities with limited computational resources to retrain the entire model. The model trained on the source data is only fine-tuned by the inner loop of the meta-learning scheme with few-shot observations. This is accompanied by a small number of gradient adaptation steps, e.g., 5 in our experiments. We compare MetaTSI with the state-of-the-art implicit time series imputation model TimeFlow.

Generalization on unseen series. We split the data used in Section 5.3 into two separate sets, each containing half of the series (about 3000 series). We pretrain the model on half of the data and fine-tune it with the meta-learning adaptation (lines 7-10 in Algorithm 1) for the other half set. Table 4 shows the result. As can be seen, MetaTSI-1D outperforms the other two models by a large margin, indicating that it learns representations to generalize rather than memorize.

Generalization on unseen cities. We then apply models to completely unseen cities that are never shown in the training set. The remaining five cities are adopted for evaluation. As new cities can have patterns different from source cities, this task is very challenging. Similarly, MetaTSI-1D still achieves desirable performance compared to its counterparts, implying great potential for cold-start problems.

TABLE 4: Generalization on unseen series and new cities.

	Models	**MetaTSI-1D**		**MetaTSI-2D**		TimeFlow	
	Cities	MAE	MSE	MAE	MSE	MAE	MSE
Unseen Series	London	**0.242**	**0.125**	0.868	1.285	0.493	0.468
	Orange	**0.239**	**0.108**	0.448	0.360	0.421	0.294
	Utrecht	**0.289**	**0.180**	0.686	0.870	0.776	1.154
	Los Angeles	**0.186**	**0.076**	0.555	0.523	0.446	0.335
	San Diego	**0.176**	**0.067**	0.552	0.522	0.468	0.365
	Riverside	**0.181**	**0.073**	0.579	0.589	0.375	0.245
	San Francisco	**0.256**	**0.132**	0.522	0.461	0.498	0.411
	Melbourne	**0.341**	**0.257**	0.754	0.955	0.439	0.374
	Bern	**0.297**	**0.179**	0.581	0.635	0.758	1.056
	Kassel	**0.336**	**0.254**	0.668	0.828	0.775	1.074
	Darmstadt	**0.410**	**0.348**	0.755	0.981	0.746	0.934
	Toronto	**0.330**	**0.238**	0.872	1.245	0.442	0.378
	Speyer	**0.454**	**0.387**	0.549	0.522	0.862	1.255
	Manchester	**0.178**	**0.082**	0.886	1.383	0.657	0.822
	Luzern	**0.338**	**0.200**	0.571	0.579	0.896	1.281
	Average	**0.284**	**0.180**	0.656	0.783	0.603	0.696
New Cities	Taipei	**0.435**	**0.375**	0.910	1.437	0.631	0.746
	Contra Costa	**0.223**	**0.098**	0.592	0.610	0.561	0.513
	Hamburg	**0.433**	**0.363**	0.776	1.035	0.740	0.997
	Munich	**0.266**	**0.189**	0.678	0.815	0.583	0.701
	Zurich	**0.416**	**0.332**	0.752	0.997	0.721	0.947
	Average	**0.354**	**0.271**	0.742	0.979	0.647	0.781

5.5 Model Efficiency and Robustness

This section details additional properties of MetaTSI, including both its computational efficiency and robustness.

5.5.1 *Inference Efficiency*

A salient property of MAML is its efficiency in adapting to different instances. We compare it with the full-training strategy when applied to all series within a single city. Figure 3 compares the training speed (time) and the parameter count for the two strategies. It is evident that meta-learning significantly reduces computational expenses and shows better scalability and parameter efficiency.

5.5.2 *Robustness with Sparser Data*

To justify the necessity of the collaborative imputation scheme developed in this paper, we evaluate the model robustness under sparse data conditions in Tables 5 and 6.

TABLE 5: Model generalization on new time series (unseen) with varying missing rates (in terms of MAE). **MetaTSI only performs meta learning procedure with a small number (3-5 steps) of adaptation steps.**

New series	San Diego			Riverside			Orange			Los Angeles		
Models	Observed rate			Observed rate			Observed rate			Observed rate		
	10%	5%	1%	10%	5%	1%	10%	5%	1%	10%	5%	1%
FourierNet†	0.254	0.274	0.565	0.264	0.268	0.501	0.212	0.339	0.589	0.254	0.271	0.492
LRTFR†	0.227	0.235	0.597	0.204	0.289	0.471	0.270	0.316	0.581	0.227	0.330	0.513
TimeFlow‡	0.389	0.505	0.759	0.320	0.424	0.695	0.479	0.661	0.876	0.391	0.568	0.764
TimeFlow†	0.403	0.468	0.782	0.390	0.398	0.726	0.340	0.513	0.856	0.404	0.487	0.802
Functa‡	0.301	0.385	0.871	0.279	0.383	0.864	0.313	0.431	0.976	0.297	0.382	0.873
Functa†	0.332	0.391	0.922	0.269	0.401	0.929	0.308	0.426	0.981	0.311	0.420	0.965
MetaTSI†	0.229	0.308	0.630	0.192	0.255	0.589	0.260	0.358	0.748	0.230	0.310	0.591
MetaTSI	**0.162**	**0.226**	**0.437**	**0.164**	**0.248**	**0.449**	**0.178**	**0.250**	**0.493**	**0.162**	**0.223**	**0.374**

†: Individual models for each city trained from scratch using the observed data. ‡: A unified model trained on the union of all data available.

TABLE 6: Model generalization on new cities (unseen) with varying missing rates (in terms of MAE). **MetaTSI only performs meta learning procedure with a small number (3-5 steps) of adaptation steps.**

New cities	Taipei			Contra Costa			Hamburg			Munich		
Models	Observed rate			Observed rate			Observed rate			Observed rate		
	10%	5%	1%	10%	5%	1%	10%	5%	1%	10%	5%	1%
SIREN+‡	0.566	0.611	0.709	0.772	0.766	0.897	0.592	0.658	0.864	0.391	0.444	0.770
TimeFlow‡	0.484	0.583	0.596	0.469	0.680	0.908	0.596	0.688	**0.742**	0.340	0.516	0.703
Functa‡	0.592	0.589	0.600	0.817	0.813	0.834	0.736	0.738	0.752	0.498	0.500	0.681
MetaTSI	**0.442**	**0.493**	**0.586**	**0.257**	**0.288**	**0.410**	**0.531**	**0.619**	0.792	**0.283**	**0.413**	**0.631**

‡: A unified model trained from scratch on the union of all data available.

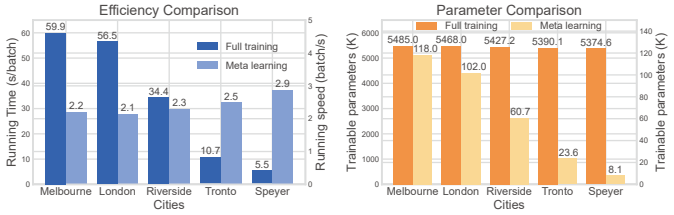


Fig. 3: Efficiency and parameter study. Note that different axes have different scales and labels for better visualization.

The pretrained MetaTSI efficiently adapts to the new tasks by exploiting the inherent knowledge in weights of the base network. The baselines are either individual models trained from scratch using limited observations or a unified model trained on all available data at the same time. Results reveal that in scenarios where the target city has very limited observations (e.g., only 1%) available, cross-city collaboration can significantly achieve performance gains.

5.5.3 *Training stability*

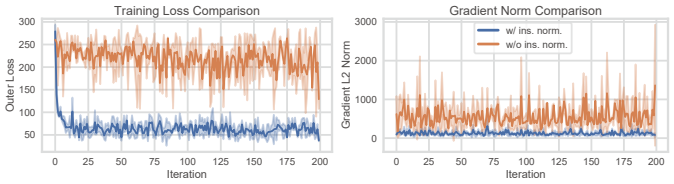


Fig. 4: Training stability using meta-learning.

Due to the inner-outer loop of MAML, the training stability can be a concern for large-scale datasets. To evaluate the potential gradient instability raised by MAML, we provide training diagnostics in Fig. 4. We record the training loss curves (top) and the ℓ-2 norm of the gradient (bottom) across iterations from multiple runs. As can be observed, the proposed masked instance normalization in Section 4.2.1 has been crucial in increasing training stability. After being equipped with it, MetaTSI shows no signs of gradient explosion or vanishing, indicating stable gradient flow and smooth convergence. These results confirm the numerical stability of our method during meta-training.

5.6 Ablation Study

To justify the rationality of each modular design, we perform architectural ablation studies in Tab. 7 using two cities. They include the following architecture variations: (1) replacing the initialization scheme of latent code; (2) replacing the hierarchical modulation scheme with a fixed phase modulation used in [42]; (3) replacing sinusoidal activation; (4) removing the meta-learning procedure; (5) removing the local loss; (6) removing the coordinate delay embedding; (7) removing the masked instance normalization; (8) removing CRF. According to the evaluation results, each modular design contributes to overall performance.

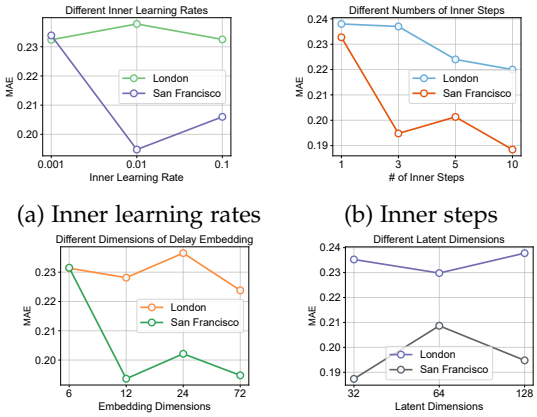
TABLE 7: Ablations studies.

	Variation	London		S.F	
		MAE	MSE	MAE	MSE
Full	MetaTSI-1D	**0.234**	**0.121**	**0.181**	**0.077**
Replace	Zero init. → Random init.	0.252	0.157	0.226	0.110
	Hierarchical → Fixed	0.318	0.223	0.814	0.989
	Sine → ReLU	0.246	0.135	0.236	0.126
w/o	Meta learning	0.289	0.199	0.312	0.200
	Variation loss	0.250	0.144	0.248	0.133
	Delay emb.	0.523	0.558	0.436	0.373
	Masked instance norm.	0.578	0.721	0.506	0.455
	CRF	0.244	0.130	0.240	0.118

5.7 Hyperparameter Study

Figure 5 examines the effects of several hyperparameters, including the number of inner steps, the inner learning rate, the dimensions of the delay embedding δ and the latent coding. There are several observations: (1) a larger inner learning rate encourages learning in data with diverse

instances; (2) a small number of inner steps with a proper dimension are enough to differentiate different instances; (3) a larger δ can boost performance, but increase complexity.



(a) Inner learning rates (b) Inner steps

(c) Dim. of delay lag (d) Dim. of latent code

Fig. 5: Hyperparameter studies.

5.8 Exploring the Latent Manifold

The meta learning procedure enables a latent code learned from data to represent the series instance. In this section, we interpret these encodings using visualization tools.

5.8.1 *t-SNE Structure.*

After auto-decode the latent code for each time series in all cities, we store these codes and project them into the two-dimensional plane using the t-SNE method. Figure 6 shows the encodings of different cities. Intriguingly, MetaTSI generally separates different cities, even without any geolocation priors. Moreover, cities with similar traffic flow patterns appear in clusters that are close to each other, e.g., cities from the bay area of CA, USA.

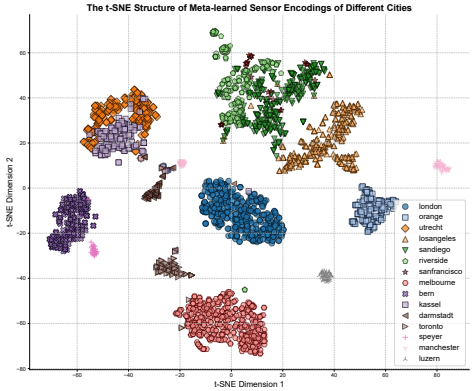


Fig. 6: The t-SNE structure of meta-learned encoding.

5.8.2 *Generating Novel Time Series.*

Recall that ones can control the output of INRs by manipulating the latent manifold. Since we have learned priors over the function space of all training samples, novel time series can be generated by interpolating the latent code. We examine the transitional behaviors reflected by the latent space by decoding an interpolated latent code $\Phi(\cdot;\phi_\mu)$ with $\phi_\mu = \mu\phi_1 + (1-\mu)\phi_2$. Figure 7 shows different generated time series (normalized) by varying μ. We observe that the interpolation path between two codes yields a "linear" transition in the time domain. This suggests that the latent space is smooth and well structured, which sheds light on the possibility of applying learned representations to generative modeling and extrapolation of adjacent sensors.

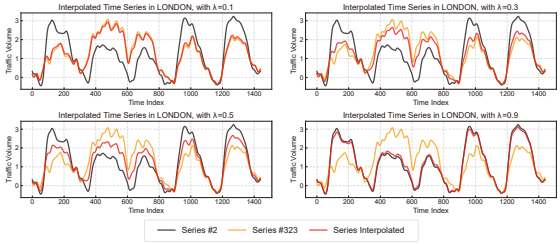


Fig. 7: Decoding a time series by interpolating latent codes.

5.8.3 *Sparse sensing with new and missing sensors*

We further evaluate MetaTSI on a spatial interpolation task, where new sensors are installed and no historical data are recorded [72]. This often occurs as real-world sensor networks are often dynamic and changing over time. In this setting, we estimate the latent code of the target sensor by averaging the latent codes of its K nearest neighboring sensors (K=3), selected based on spatial proximity. This approach is motivated by the findings in section 5.8.2 that the learned latent space exhibits approximate linearity and preserves spatial correlations. Leveraging the similarity among neighboring sensors in the latent space, we can infer the representation of a new sensor. Once the approximate latent code is obtained, we directly decode it using the trained MetaTSI decoder without any additional gradient-based updates. This allows the model to make fast and effective predictions for unseen sensor locations. The experimental results in Fig. 8 demonstrate that this strategy achieves promising performance, validating the potential of MetaTSI to handle spatial dynamics in real-world sensor networks.

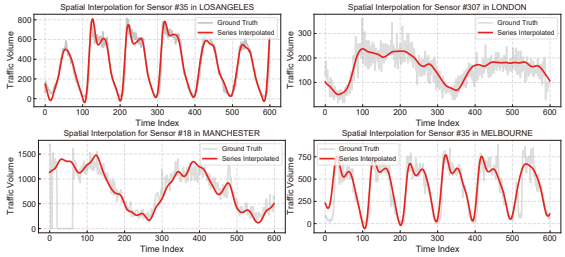
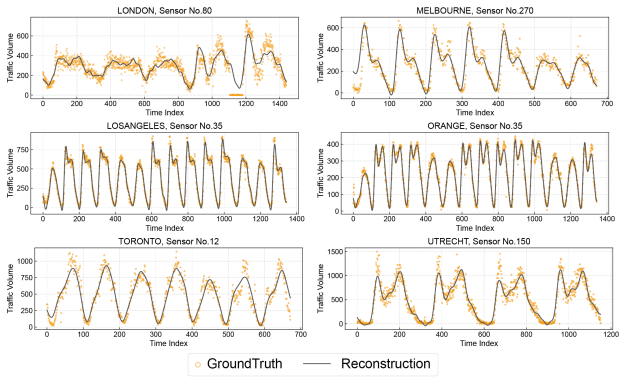


Fig. 8: Spatial interpolation using aggregated latent codes.

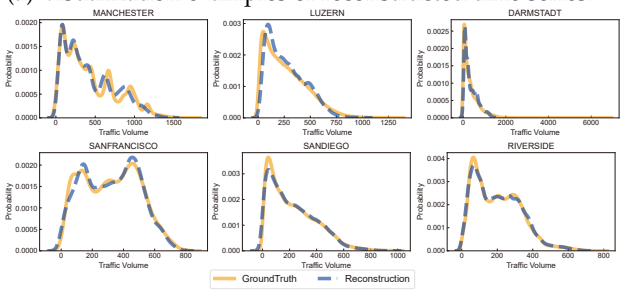
5.9 Case Study

5.9.1 *Imputation Visualization*

Figure 9 (a) displays examples of the time series imputation results in different cities and (b) shows the histogram of the true and predicted values (probability). These plots clearly show that although different cities feature distinct traffic flow observations, the shared patterns make transfer between cities possible. Our model captures these common structures and accurately reconstructs the ground truth.



(a) Visualization examples of reconstructed time series.



(b) Visualization examples of reconstructed distribution.

Fig. 9: Visualization examples of imputation results.

5.9.2 *Frequency Analysis*

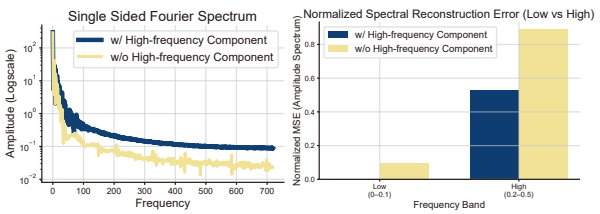


Fig. 10: Frequency analysis of the imputation results.

To highlight the importance of high-frequency features, we compare the frequency responses and band reconstruction accuracies of models with and without high-frequency features in Fig. 10. First, the figure on the left shows the frequency response of different models combining low- and high-frequency components. Compared to the baseline, MetaTSI retains significantly more spectral energy at higher frequencies, demonstrating its capacity to mitigate spectral bias. Second, the right figure shows normalized reconstruction errors across low- and high-frequency bands. This shows a direct performance comparison across frequency scales. MetaTSI consistently achieves lower relative error, especially in the high-frequency range.

6 Conclusion and Future Direction

In this study, we introduce a novel time series imputation framework leveraging meta-learning implicit neural representations called MetaTSI. Thanks to the generic architecture, MetaTSI originally pretrained to distinguish different instances can then generalize to heterogeneous cities to achieve collaborative imputation tasks that vary in spatiotemporal resolutions and observation patterns. Experimental results indicate that MetaTSI not only achieves superior imputation accuracy but also excels in generalizability across diverse tasks. In addition, MetaTSI recovers the global structure of the signal manifold, allowing easy interpolation between nearby signals. Although promising results are presented, it revealed some limitations that need future efforts. First, while meta-learning facilitates efficient adaptation of new samples, pretraining it in the source data is computationally expensive and less stable than standard supervised learning. Second, this work only focuses on a single data modality. Extending MetaTSI to diverse modalities of urban data requires more sophisticated architectures. Third, recent advances in large models have shown strong potential in general-purpose urban prediction tasks. They use pretrained language models as the backbone of forecasting models to generate predictions and as spatiotemporal embedders to process information [15], [73], [74], [75]. Although our work focuses on lightweight and continuous function modeling suited to sparse and resource-limited scenarios, future directions could explore hybrid approaches that integrate INRs with spatiotemporal foundation models to further enhance adaptability and semantic reasoning.

Acknowledgments

This research was partially sponsored by the National Natural Science Foundation of China (524B2164, 52125208), partially conducted at the Research Institute for Artificial Intelligence of Things (RIAIoT) at PolyU and funded by Hong Kong Research Grants Council (RGC) under the Theme-based Research Scheme with grant No. T41-603/20-R, and partially sponsored by grants from the RGC with project Nos. PolyU/15206322 and PolyU/15227424.

References

[1] Z. Fan, F. Zhang, B. P. Loo, and C. Ratti, "Urban visual intelligence: Uncovering hidden city profiles with street view images," *Proceedings of the National Academy of Sciences*, vol. 120, no. 27, p. e2220417120, 2023.

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

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

[4] 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.

[5] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," *Advances in neural information processing systems*, vol. 31, 2018.

[6] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[7] M. Liu, H. Huang, H. Feng, L. Sun, B. Du, and Y. Fu, "Pristi: A conditional diffusion framework for spatiotemporal imputation," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 1927–1939.

[8] Z. Senane, L. Cao, V. L. Buchner, Y. Tashiro, L. You, P. A. Herman, M. Nordahl, R. Tu, and V. Von Ehrenheim, "Self-supervised learning of time series representation via diffusion process and imputation-interpolation-forecasting mask," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2560–2571.

[9] W. Du, D. Côté, and Y. Liu, "Saits: Self-attention-based imputation for time series," *Expert Systems with Applications*, vol. 219, p. 119619, 2023.

[10] T. Nie, G. Qin, W. Ma, Y. Mei, and J. Sun, "Imputeformer: Low rankness-induced transformers for generalizable spatiotemporal imputation," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2260–2271.

[11] L. Han, H.-J. Ye, and D.-C. Zhan, "The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[12] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," *Advances in neural information processing systems*, vol. 29, 2016.

[13] X. Chen, Z. Cheng, H. Cai, N. Saunier, and L. Sun, "Laplacian convolutional representation for traffic time series imputation," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[14] M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 7, pp. 1816–1825, 2016.

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

[16] Q. Ma, Z. Liu, Z. Zheng, Z. Huang, S. Zhu, Z. Yu, and J. T. Kwok, "A survey on time-series pre-trained models," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[17] W. Ma and G. H. Chen, "Missing not at random in matrix completion: The effectiveness of estimating missingness probabilities under a low nuclear norm assumption," *Advances in neural information processing systems*, vol. 32, 2019.

[18] J. Xue, N. Jiang, S. Liang, Q. Pang, T. Yabe, S. V. Ukkusuri, and J. Ma, "Quantifying the spatial homogeneity of urban road networks via graph neural networks," *Nature Machine Intelligence*, vol. 4, no. 3, pp. 246–257, 2022.

[19] Z. Shao, F. Wang, Y. Xu, W. Wei, C. Yu, Z. Zhang, D. Yao, T. Sun, G. Jin, X. Cao, G. Cong, C. S. Jensen, and X. Cheng, " Exploring Progress in Multivariate Time Series Forecasting: Comprehensive Benchmarking and Heterogeneity Analysis ," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, Oct. 2024.

[20] Y. Du, K. Collins, J. Tenenbaum, and V. Sitzmann, "Learning signal-agnostic manifolds of neural fields," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8320–8331, 2021.

[21] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.

[22] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in neural information processing systems*, vol. 33, pp. 7537–7547, 2020.

[23] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[24] Y. Chen, S. Liu, and X. Wang, "Learning continuous image representation with local implicit image function," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8628–8638.

[25] Y. Luo, X. Zhao, Z. Li, M. K. Ng, and D. Meng, "Low-rank tensor function representation for multi-dimensional data recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[26] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Learning deep time-index models for time series forecasting," in *International Conference on Machine Learning*. PMLR, 2023, pp. 37 217–37 237.

[27] T. Nie, G. Qin, W. Ma, and J. Sun, "Spatiotemporal implicit neural representation as a generalized traffic data learner," *arXiv preprint arXiv:2405.03185*, 2024.

[28] H. Whitney, "Differentiable manifolds," *Annals of Mathematics*, vol. 37, no. 3, pp. 645–680, 1936.

[29] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*. Springer, 2006, pp. 366–381.

[30] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[31] X. Miao, Y. Wu, L. Chen, Y. Gao, and J. Yin, "An experimental survey of missing data imputation algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6630–6650, 2022.

[32] X. Ma, Z. Wu, M. Ma, M. Zhao, F. Yang, Z. Du, and W. Zhang, "Steinformer: Spatial-temporal interaction transformer architecture for remote sensing change detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[33] D. Liu, Y. Wang, C. Liu, X. Yuan, K. Wang, and C. Yang, "Scope-free global multi-condition-aware industrial missing data imputation framework via diffusion transformer," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[34] N. Karmitsa, S. Taheri, A. Bagirov, and P. Mäkinen, "Missing value imputation via clusterwise linear regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1889–1901, 2020.

[35] Y. Gong, Z. Li, J. Zhang, W. Liu, Y. Yin, and Y. Zheng, "Missing value imputation for multi-view urban statistical data via spatial correlation learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 686–698, 2021.

[36] T. Nie, G. Qin, and J. Sun, "Truncated tensor schatten p-norm based approach for spatiotemporal traffic data imputation with complicated missing patterns," *Transportation Research Part C: Emerging Technologies*, vol. 141, p. 103737, 2022.

[37] A. Blázquez-García, K. Wickstrøm, S. Yu, K. Ø. Mikalsen, A. Boubekki, A. Conde, U. Mori, R. Jenssen, and J. A. Lozano, "Selective imputation for multivariate time series datasets with missing values," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 9490–9501, 2023.

[38] X. Wang, Y. Wu, D. Zhuang, and L. Sun, "Low-rank hankel tensor completion for traffic speed estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 4862–4871, 2023.

[39] S. N. Shukla and B. Marlin, "Interpolation-prediction networks for irregularly sampled time series," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=r1efr3C9Ym

[40] S. C.-X. Li and B. Marlin, "Learning from irregularly-sampled time series: A missing data perspective," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5937–5946.

[41] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," *Neurocomputing*, vol. 441, pp. 161–178, 2021.

[42] E. L. Naour, L. Serrano, L. Migus, Y. Yin, G. Agoua, N. Baskiotis, V. Guigue *et al.*, "Time series continuous modeling for imputation and forecasting with implicit neural representations," *arXiv preprint arXiv:2306.05880*, 2023.

[43] E. Dupont, A. Goliński, M. Alizadeh, Y. W. Teh, and A. Doucet, "Coin: Compression with implicit neural representations," *arXiv preprint arXiv:2103.03123*, 2021.

[44] X. Luo, W. Xu, Y. Ren, S. Yoo, and B. Nadiga, "Continuous field reconstruction from sparse observations with implicit neural networks," *arXiv preprint arXiv:2401.11611*, 2024.

[45] E. Fons, A. Sztrajman, Y. El-Laham, A. Iosifidis, and S. Vyetrenko, "Hypertime: Implicit neural representation for time series," *arXiv preprint arXiv:2208.05836*, 2022.

[46] C. Kim, D. Lee, S. Kim, M. Cho, and W.-S. Han, "Generalizable implicit neural representations via instance pattern composers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 808–11 817.

[47] E. Dupont, H. Kim, S. A. Eslami, D. J. Rezende, and D. Rosenbaum, "From data to functa: Your data point is a function and you can treat it like one," in *International Conference on Machine Learning*. PMLR, 2022, pp. 5694–5725.

[48] V. Sitzmann, E. Chan, R. Tucker, N. Snavely, and G. Wetzstein, "Metasdf: Meta-learning signed distance functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 136–10 147, 2020.

[49] I. Mehta, M. Gharbi, C. Barnes, E. Shechtman, R. Ramamoorthi, and M. Chandraker, "Modulated periodic activations for generalizable local functional representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 214–14 223.

[50] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in *Proceedings of the*

28th International Joint Conference on Artificial Intelligence, 2019, pp. 1893–1899.
[51] Y. Tang, A. Qu, A. H. Chow, W. H. Lam, S. C. Wong, and W. Ma, "Domain adversarial spatial-temporal network: A transferable framework for short-term traffic forecasting across cities," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1905–1915.
[52] X. Ouyang, Y. Yang, Y. Zhang, W. Zhou, J. Wan, and S. Du, "Domain adversarial graph neural network with cross-city graph structure learning for traffic prediction," *Knowledge-Based Systems*, vol. 278, p. 110885, 2023.
[53] Y. Jin, K. Chen, and Q. Yang, "Transferable graph structure learning for graph-based traffic forecasting across cities," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 1032–1043.
[54] X. Ouyang, Y. Yang, W. Zhou, Y. Zhang, H. Wang, and W. Huang, "Citytrans: Domain-adversarial training with knowledge transfer for spatio-temporal prediction across cities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 1, pp. 62–76, 2023.
[55] Y. Zhang, H. Lu, N. Liu, Y. Xu, Q. Li, and L. Cui, "Personalized federated learning for cross-city traffic prediction," in *33rd International Joint Conference on Artificial Intelligence, IJCAI 2024*. International Joint Conferences on Artificial Intelligence, 2024, pp. 5526–5534.
[56] Y. Wang, T. Zheng, Y. Liang, S. Liu, and M. Song, "Cola: Cross-city mobility transformer for human trajectory simulation," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 3509–3520.
[57] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in *The world wide web conference*, 2019, pp. 2181–2191.
[58] B. Lu, X. Gan, W. Zhang, H. Yao, L. Fu, and X. Wang, "Spatio-temporal graph few-shot learning with cross-city knowledge transfer," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1162–1172.
[59] Y. Jin, K. Chen, and Q. Yang, "Selective cross-city transfer learning for traffic prediction via source city region re-weighting," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 731–741.
[60] Y. Zhang, Y. Li, X. Zhou, and J. Luo, "Mest-gan: Cross-city urban traffic estimation with me ta s patial-t emporal g enerative a dversarial n etworks," in *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2022, pp. 733–742.
[61] Z. Liu, G. Zheng, and Y. Yu, "Cross-city few-shot traffic forecasting via traffic pattern bank," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 1451–1460.
[62] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of statistical Physics*, vol. 65, pp. 579–616, 1991.
[63] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Physical review letters*, vol. 45, no. 9, p. 712, 1980.
[64] T. Wu, X. Gao, F. An, X. Sun, H. An, Z. Su, S. Gupta, J. Gao, and J. Kurths, "Predicting multiple observations in complex systems through low-dimensional embeddings," *Nature Communications*, vol. 15, no. 1, p. 2242, 2024.
[65] T. Nie, Y. Mei, G. Qin, J. Sun, and W. Ma, "Channel-aware low-rank adaptation in time series forecasting," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 3959–3963.
[66] R. Fathony, A. K. Sahu, D. Willmott, and J. Z. Kolter, "Multiplicative filter networks," in *International Conference on Learning Representations*, 2020.
[67] D. Lee, C. Kim, M. Cho, and W. S. HAN, "Locality-aware generalizable implicit neural representation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[68] X. Ma, Z. Ni, and X. Chen, "Tinyvim: Frequency decoupling for tiny hybrid vision mamba," *arXiv preprint arXiv:2411.17473*, 2024.
[69] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," *Advances in neural information processing systems*, vol. 31, 2018.
[70] S. Liu, X. Li, G. Cong, Y. Chen, and Y. Jiang, "Multivariate time-series imputation with disentangled temporal representations," in *The Eleventh international conference on learning representations*, 2023.
[71] S. N. Shukla and B. M. Marlin, "Multi-time attention networks for irregularly sampled time series," *arXiv preprint arXiv:2101.10318*, 2021.
[72] T. Nie, G. Qin, Y. Wang, and J. Sun, "Correlating sparse sensing for large-scale traffic speed estimation: A laplacian-enhanced low-rank tensor kriging approach," *Transportation Research Part C: Emerging Technologies*, vol. 152, p. 104190, 2023.
[73] W. Li, D. Yao, R. Zhao, W. Chen, Z. Xu, C. Luo, C. Gong, Q. Jing, H. Tan, and J. Bi, "Stbench: Assessing the ability of large language models in spatio-temporal analysis," *arXiv preprint arXiv:2406.19065*, 2024.
[74] Z. Li, L. Xia, J. Tang, Y. Xu, L. Shi, L. Xia, D. Yin, and C. Huang, "Urbangpt: Spatio-temporal large language models," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5351–5362.
[75] J. He, T. Nie, and W. Ma, "Geolocation representation from large language models are generic enhancers for spatio-temporal learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 16, 2025, pp. 17 094–17 104.

Tong Nie (Student Member, IEEE) received the B.S. degree from the college of civil engineering, Tongji University, Shanghai, China. He is currently pursuing dual Ph.D. degrees with Tongji University and The Hong Kong Polytechnic University. He has published several papers in top-tier venues in the field of spatiotemporal data modeling, including KDD, AAAI, CIKM, IEEE TKDE/TII/TITS, and TR-Part C/E. His research interests include spatiotemporal learning, time series analysis, and large language models. His research is funded by the National Natural Science Foundation of China.

Wei Ma (Member, IEEE) received the bachelor's degree in civil engineering and mathematics from Tsinghua University, China, and the master's degree in machine learning and civil and environmental engineering and the Ph.D. degree in civil and environmental engineering from Carnegie Mellon University, USA. He is currently an Assistant Professor with the Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University (PolyU). His current research interests include machine learning, data mining, and transportation network modeling, with applications for smart and sustainable mobility systems.

Jian Sun received the Ph.D. degree in transportation engineering from Tongji University, Shanghai, China. He is currently a Professor of transportation engineering with Tongji University. He has published more than 100 papers in SCI journals. His research interests include intelligent transportation systems, traffic flow theory, AI in transportation, and traffic simulation.

Yu Yang is currently an Assistant Professor with the Centre for Learning, Teaching, and Technology, The Education University of Hong Kong. He received the Ph.D. degree in Computer Science from The Hong Kong Polytechnic University in 2021. His research interests include spatiotemporal data analysis, representation learning, urban computing, and learning analytics.

Jiannong Cao (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in computer science from Washington State University, Pullman, WA, USA, in 1986 and 1990, respectively. He is currently the Chair Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests include parallel and distributed computing, mobile computing, and big data analytics. Dr. Cao has served as a member of the Editorial Boards of several international journals, a Reviewer for international journals/conference proceedings, and also as an Organizing/ Program Committee member for many international conferences.