A Diffusion Model for Traffic Data Imputation

Bo Lu, Qinghai Miao, *Senior Member, IEEE*, Yahui Liu, Tariku Sinshaw Tamir, Hongxia Zhao, *Senior Member, IEEE*, Xiqiao Zhang, Yisheng Lv, *Senior Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—Imputation of missing data has long been an important topic and an essential application for intelligent transportation systems (ITS) in the real world. As a state-of-the-art generative model, the diffusion model has proven highly successful in image generation, speech generation, time series modelling etc. and now opens a new avenue for traffic data imputation. In this paper, we propose a conditional diffusion model, called the implicit-explicit diffusion model, for traffic data imputation. This model exploits both the implicit and explicit feature of the data simultaneously. More specifically, we design two types of feature extraction modules, one to capture the implicit dependencies hidden in the raw data at multiple time scales and the other to obtain the long-term temporal dependencies of the time series. This approach not only inherits the advantages of the diffusion model for estimating missing data, but also takes into account the multiscale correlation inherent in traffic data. To illustrate the performance of the model, extensive experiments are conducted on three real-world time series datasets using different missing rates. The experimental results demonstrate that the model improves imputation accuracy and generalization capability.**

***Index Terms*—Data imputation, diffusion model, implicit feature, time series, traffic data.**

I. Introduction

THE quality of traffic data is the basis for the planning and scheduling of intelligent transportation systems (ITS). However, in real-world traffic scenarios, the problem of missing data due to traffic sensor malfunctions, transmission failures, etc. is inevitable, which seriously affects the ability of ITS to formulate efficient and appropriate management plans. The average rate of missing data in the performance measurement system (PeMS) in Los Angeles is 15%. In Beijing, China, this figure is as high as about 25% for some loop traffic detectors [1]. Manually imputing missing data is a laborious and ongoing process that can not be completed once and for all. Therefore, many approaches have been studied in the last decades to solve the problem of missing data accurately and efficiently. They fall into two categories, those based on statistics and those based on machine learning. In statistical methods, the fastest and easiest way is to impute the missing data with special values [2], such as the previous value, the next value, the mean value, the median value, and the mode value. However, these methods are inherently linear and are therefore unsuitable for dealing with high-dimensional, non-stationary, and complex traffic data. In addition to traditional statistical approaches, machine learning-based approaches are gaining popularity in a variety of intelligent transportation systems (ITS) tasks [3]–[7], among which deep learning techniques are the most prominent. Deep learning techniques possess incredible ability in modeling complex and nonlinear data patterns [8] as well as extracting inherent features in high-dimensional data. They have demonstrated good performance in restoring the state of the traffic data and improving imputation.

In this paper, we propose a novel deep learning-based method for traffic data imputation. We address the limitations of common compensation methods by incorporating more observable implicit information, enabling our model to make better compensations. Furthermore, we enhance the modeling of traffic time series data by fusing implicit features from different temporal granularities with explicit features. The diffusion model is used to fully exploit the implicit and explicit correlation to improve the imputation accuracy. In addition, the multiscale implicit features are considered in the modeling at multi-temporal granularities. It shows that the proposed method for traffic data imputation has superior performance. The contributions of our work are threefold:

1) We propose a conditional diffusion model-based approach, termed the implicit-explicit diffusion model, for traffic data imputation, in which an individual implicit extraction module is developed to capture the hidden multiscale implicit dependencies of raw traffic data. The diffusion model acts as a generator to infer the missing data from random noise.

2) We improve the modeling of traffic time series data by

Manuscript received March 3, 2024; revised April 17, 2024; accepted May 31, 2024. This work was partially supported by the National Natural Science Foundation of China (62271485), and the SDHS Science and Technology Project (HS2023B044). Recommended by Associate Editor Xiaohua Ge. *(Corresponding authors: Hongxia Zhao and Xiqiao Zhang.)*

Citation: B. Lu, Q. Miao, Y. Liu, T. Tamir, H. Zhao, X. Zhang, Y. Lv, and F.-Y. Wang, "A diffusion model for traffic data imputation," *IEEE/CAA J. Autom. Sinica*, vol. 12, no. 3, pp. 606–617, Mar. 2025.

B. Lu was with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China. He is now with Baidu Inc., Beijing 100190, China (e-mail: lubo21@mails.ucas.ac.cn).

Q. Miao is with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: miaoqh@ucas.ac.cn).

Y. Liu was with the State Key Laboratory for Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. She is now with Meituan, Beijing 100050, China (e-mail: liuyh@stu.xmu.edu.cn).

T. Tamir was with the State Key Laboratory for Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. He is now with Guangdong University of Technology, Guangzhou 510520, China (e-mail: Tamir@gdut.edu.cn).

H. Zhao, Y. Lv, and F.-Y. Wang are with the State Key Laboratory for Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: hongxia.zhao@ia.ac.cn; yisheng.lv@ia.ac.cn; feiyue.wang@ia.ac.cn).

X. Zhang is with the Department of Transportation Engineering, School of Transportation Science and Technology, Harbin Institute of Technology, Harbin 150090, China (e-mail: zxqjuly@hit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2024.124611

fusing implicit features of different temporal granularities with explicit features, resulting in more sophisticated and accurate imputation.

3) We conducted comprehensive empirical evaluations on diverse time series datasets with varying degrees of missing data to assess the performance of the proposed implicit-explicit diffusion model. The results consistently showcased the model's superiority over the majority of existing imputation methods in terms of imputation accuracy.

The rest of this paper is organized as follows. Section II reviews the related works on time series imputation. In Section III, we introduce the notations, preliminaries, and formulation of the task, and explain the imputation process in detail. In Section IV, we evaluate the performance of the proposed algorithm and discuss the experimental results. Finally, the conclusion and future work are described in Section V.

II. Literature Review

Data is the foundation for the planning and control of ITS [9], [10]. In the past decades, researchers have developed a wide range of methodologies and theories for data imputation. In this section, the existing studies on data imputation are classified into two main categories: Statistical and machine learning-based.

Statistical methods are frequently used for imputing time series with low computational complexity, which is more applicable in many scenarios that do not require high accuracy. The last observation carried forward (LOCF) [2] method, one of the simplest statistical methods, selects adjacent position values to impute the data using the previous position of the missing point. In contrast, the next observation carried back (NOCB) method works in the opposite direction, taking the first observation after the missing value and carrying it backward. However, the unpredictable fluctuations of real time series data often lead to suboptimal results when the direct insect approach is used. Therefore, other traditional methods of statistical imputation involve substituting the missing data with the statistics [2] (such as mean, median, and mode values) or the most similar one from the dataset. Besides, the KNN imputation approach [11] selects the most similar values from the dataset to impute missing data. The statistical learning-based method is another classification of statistical methods, which includes matrix-based [12] and tensor-based approaches. Among the tensor-based imputation fields, the low-rank tensor completion methods [13], [14] and the tensor decomposition approach [15] have made progress in the last decade.

Machine learning-based approaches include tree-based, regression-based, and deep learning-based methods. The tree-based imputation method builds one decision tree model for each incomplete feature that contains missing values, such as XGBI [16]. The regression-based method is effective for imputing the missing data, including linear regression, iterative, quadratic, and autoregression [17], such as ARMA, ARIMA and the multivariate imputation [18] by chained equations (MICE) are classic regression-based models. However, these methods do not consider the random fluctuation of traffic data. With the development of computing power and public datasets, deep learning-based techniques have demonstrated impressive performance in the field of imputation. Deep networks are able to capture high-dimensional, complex, and dynamic traffic features [19]–[22]. Lv *et al.* [23] applied deep architecture models for the first time in 2015, using stacked autoencoders (SAEs) as building blocks to represent traffic time series features. Based on SAEs, Duan *et al.* [24] viewed traffic data imputation as the process of missing data recovery. They replaced the middle layers of SAEs with a denoising autoencoder and proposed a denoising stacked autoencoder. Missing data is not only limited to the field of transportation [25]–[29], it is also a prevalent issue in medical datasets, airspace complexity prediction, and others. Yoon *et al.* [30] reconstructed missing data by leveraging the correlations within data streams and across data streams, as reported in their paper.

Since time series capturing nodes are located in diverse geographic areas, spatial characteristics are often obscured within them. Graph convolution networks are building blocks for learning graph-structured data [31]. Wu *et al.* [32] proposed graph WaveNet, which captures spatial-temporal dependencies by combining graph convolution with dilated causal convolution. NAOMI [33] focuses on spatial dependency, which takes advantage of the spatio-temporal multiresolution structure [34]. It decodes from coarse to fine-grained resolutions recursively using a divide-and-conquer strategy. Cao *et al.* [35] built a bidirectional recurrent dynamical system (BRITS) in 2018, which estimates missing values directly without any specific assumptions. Both BRITS and NAOMI adopt bidirectional RNN structures. Besides spatial features, temporal dependence is also critical. GLAMA [36] was proposed in 2020 to capture global and local features in time series with multi directional attention learning. Wei *et al.* [37] designed a novel spatial-temporal graph synchronous aggregation model for traffic prediction. Researchers try to impute the missing data with more efficient methods, such as self-attention mechanism [38], [39], residual layer [40], and structure state space models (SSSD) [41].

Data imputation as data generation. The success of generative adversarial nets (GAN) [42] in data generation provided a novel way to impute missing data. Luo *et al.* [43] treat the problem of missing value imputation as data generation. They incorporated a modified gate recurrent unit into GAN so that the imputer could generate reasonable value from noise. GANs have become popular in recent years because they produce high quality results [44], and many GAN-based imputation approaches [45]–[47] have emerged. However, GANs [48] have a reputation for being difficult to train. Diffusion models have been developed to address the GAN training convergence. Several diffusion-based generative models, such as diffusion probabilistic models [49], noise-conditioned score networks [50], and denoising diffusion probabilistic models (DDPM) [51], have been proposed with similar basic ideas. DiffWave [52] ignites diffusion models for time series problems. Tashiro *et al.* [40] proposed a conditional score-based diffusion model (CSDI) for time series imputation based on DiffWave architecture. Alcaraz and Strodthoff [41] proposed an imputation method, which combines conditional diffusion models as a generator and structured state space models as a tool to capture long-term dependencies in time series. SSSD and CSDI have a similar network structure. They are designed based on DiffWave architecture. Our implicit-explicit diffu-

sion model employs a DiffWave-based framework and incorporates both implicit and explicit modules to model implicit and explicit features, respectively.

III. Methodology

Traffic time series data, often referred to as temporal data, represents a sequential arrangement of data points with temporal indexing. Fig. 1 provides a visual depiction of the data processing flow for traffic time series data. Suppose we have a complete set of traffic time series data, existing in the form of a $L \times K$ matrix. We can generate incomplete traffic data observations from a realistic traffic scenario using an artificial mask matrix. In this incomplete traffic data, observed values are represented by blue cells, and missing values are indicated by blank cells. The incomplete traffic data and the generated mask matrix (indicating observed and missing locations) are then fed as inputs into a model. The model's output is complete time series data, including the original observed values and the imputed or predicted values for the missing entries. The figure visually depicts the entire process of handling missing data in traffic time series, encompassing mask generation, model input and output, and the final imputation of missing values.

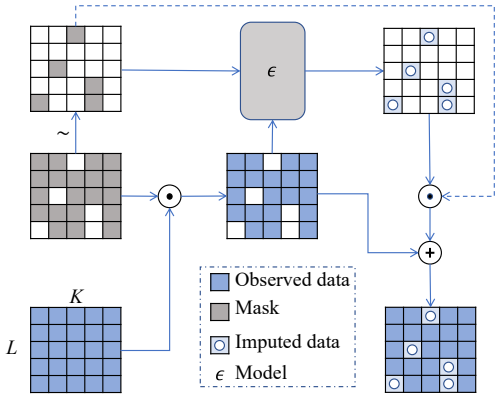


Fig. 1. Data processing flow, including raw data acquisition, mask generation, model input and output, and a final illustration of imputation.

We express the state of each time series as $\mathbf{Z} = x_{1:L,1:K} \in \mathbb{R}^{L \times K}$, where K is the number of features, and L is the length of the time series. Thus, the features of each batch with temporal interval l and spatial interval τ can be represented as follows:

$$Z_t^{(k)} = \begin{bmatrix} X_t^k & \cdots & X_t^{k+\tau-1} & X_t^{k+\tau} \\ \vdots & \ddots & \vdots & \vdots \\ X_{t+l}^k & \cdots & X_{t+l}^{k+\tau-1} & X_{t+l}^{k+\tau} \end{bmatrix}. \tag{1}$$

In our paper, we categorize feature dimensions based on the number of traffic stations, ensuring numerical consistency between them. Let X_t^k denote the value in a time series, representing the values of station k at time t. We hereby introduce the concept of a mask matrix, denoted as $\mathbf{M} \in \{0,1\}^{L \times K}$, which is instrumental in delineating the status of data points as either observed or missing. Herein, K corresponds to the total number of features under imputation, while L represents the total number of observations within the dataset. The elements of the mask matrix are defined as follows:

$$M_t^{(k)} = \begin{bmatrix} m_t^k & \cdots & m_t^{k+\tau-1} & m_t^{k+\tau} \\ \vdots & \ddots & \vdots & \vdots \\ m_{t+l}^k & \cdots & m_{t+l}^{k+\tau-1} & m_{t+l}^{k+\tau} \end{bmatrix}. \tag{2}$$

The mask variable is denoted as

$$m_t^{(k)} = \begin{cases} 1, & \text{missing} \\ 0, & \text{observed}. \end{cases} \tag{3}$$

Therefore, the representation of missing data is given by $\widetilde{Z} = Z_t^{(k)} \cdot M_t^{(k)}$, where " $\cdot$ " denotes element-wise multiplication. For the avoidance of doubt and to ensure clarity, the definitions of the notations used in this paper are provided in Table I.

TABLE I
NOTATIONS AND EXPLANATIONS

Notation	Explanation
Z	Original time series data
Z_θ	Time series output from model inference
$\widetilde{Z}$	Missing time series data
M	Mask of time series data
$\mathcal{F}_{imp}$	Multiscale feature set
K	Number of traffic sensor nodes
L	Length of time series
N	Number of feature nodes
C	Number of feature channels
P	Padding size for dilated causal convolution
T_emb	Embedding for diffusion steps

Figure2.central

A. Imputation With Implicit-Explicit Diffusion Model

The architecture of the implicit-explicit diffusion model is illustrated in Fig. 2. The input sequence is decomposed by the DETACH module into noise $\mathbf{X}$, observations data, mask $\mathbf{M}$, which all have the same dimension with a size of $[N, C, L]$ and implicit information $\mathcal{F}_{imp}$, with a size of $[1, C, L]$. The model first decomposes the input sequence into noise, observed data, and a mask, while also extracting implicit information. It then processes the noise, observed data, mask, and implicit information through separate implicit extraction and explicit extraction modules. The outputs of these two modules are combined and passed through residual convolutional layers, along with convolutional operations and diffusion step embedding, to produce the final output, effectively modeling and imputing missing values in the time series data.

B. Data Preprocessing and Notions

Note that we set N and L by ourselves in this case. C depends on how many granularities of implicit information are added to the model learning, and L is consistent with the division of the time series. The explicit and implicit information is then sent to the corresponding modules for feature extraction.

The design of the implicit extraction modules is inspired by time convolutional neural network (TCN) [53], and involves performing an dilated causal convolution on the input received

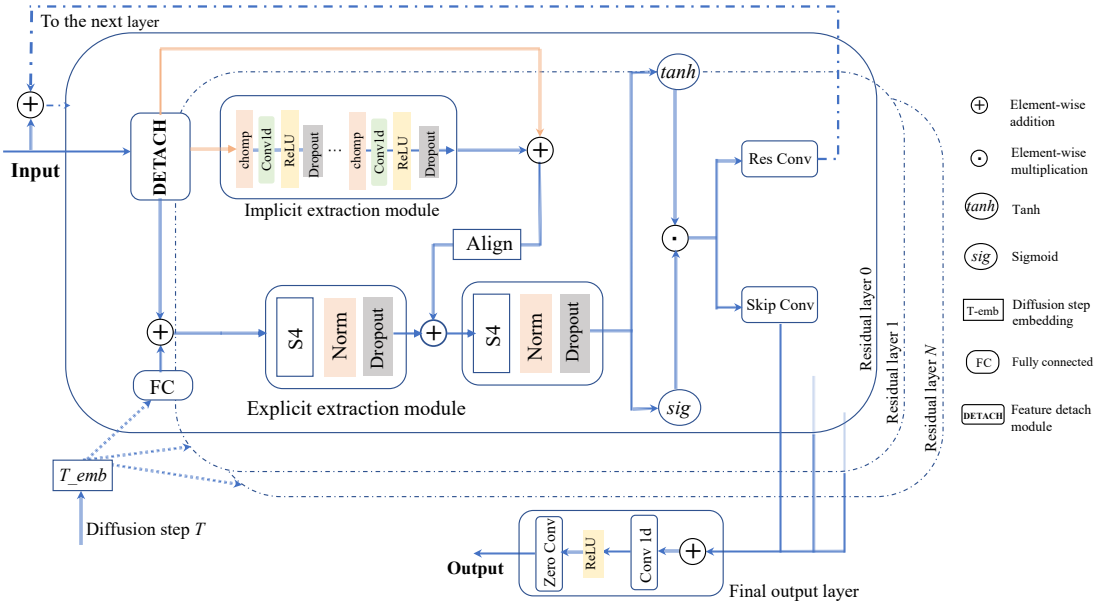


Fig. 2. The framework of the proposed implicit-explicit diffusion model.

by the module. The size of the output after convolution is actually $[1, C, L + P]$ (where P is the number of padding in the dilated convolution). To keep the size of input and the output the same, it is also necessary to trim the tensor by cutting off the extra padding part to ensure that the time step of the output is L. Then, the rectified linear units (ReLU) are added next to the chomp layer to increase the nonlinear interaction between the layers. The last dropout is to add a probabilistic process to each neuron in each layer on top of the normal neural network to randomly remove some neurons to avoid overfitting. By stacking the dilated convolutional layer with different dilation factors [54], we can obtain the long-term and short-term time series trends. The output of the implicit feature extraction module is then matched using the align module so that its dimension becomes $[1, C, L]$, which matches the original features and facilitates feature fusion. We process the original time series data using the explicit feature module. We embed a diffusion step t into a higher dimensional space, such as 128 dimensions, and then add the explicit feature with the diffusion step embedding T_emb. This is necessary because the model needs to output distinct $\theta(\cdot, t)$ for different step t. Based on works of [52] and [55], we use the following step embedding method:

$$T\_emb = (\sin(t \times 10^{\frac{0\times4}{63}} t), \ldots, \sin(t \times 10^{\frac{63\times4}{63}} t) \cos(t \times 10^{\frac{0\times4}{63}} t), \ldots, \cos(t \times 10^{\frac{63\times4}{63}} t)). \quad (4)$$

We feed the summed result into the first explicit extraction module, where the state space model captures the long-term dependencies in the time series. Then, we pass the output of the first explicit extraction module to the second implicit extraction module after fusing it with the alignment result. After performing skip convolution and residual convolution on the output of the second explicit extraction module, the output of the residual convolution is added to the input as input for the next residual layer. The skip connections from each residual layer are combined as input for the final output layer. The result of the final output layer is denoted as Z_θ. From (1) and (2), the final imputation result Z_{imputed} can be calculated as follows:

$$Z_{\text{imputed}} = Z_\theta \cdot M_t^{(k)} \cdot Z_t^{(k)} \cdot (1 - M_t^{(k)}). \quad (5)$$

C. Diffusion Probabilistic Generative Model

The diffusion probabilistic generative model is one of the generative models and has shown excellent power in the field of image generation [56]. The diffusion model has forward and reverse processes, where the forward process denotes a diffusion process. During the forward process, Gaussian random noise is gradually added to the sample, and in the reverse process, the original sample features are restored by denoising. In our research, we consider the reverse process for generating time series missing data. Given a data point sampled from the real time series data distribution $x_0 \sim q(x)$. Before starting the noise addition process, we define a total step T and then add noise to the samples in T steps, it should be noticed that each step of noise addition and denoising is based on the previous step. The diffusion process belongs to a Markov process, where the next step x_t is determined by adding noise to the current step x_{t-1}. Additionally, the diffusion step length at each step is controlled by $\beta_t \in (0, 1)$.

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}). \quad (6)$$

In DDPM [51], the researchers set the variance manually as $\{\beta_t\}_{t=1}^T$ used for each step of the noise. During the diffusion process, we take the value of the noise added at each step according to the set variance schedule.

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \quad (7)$$

We set the variance schedule in advance so that we can directly sample $x_t \sim q(x_t|x_0)$ based on x_0 at each step. The reverse process is denoising, which is a process of continu-

ously applying x_{t-1} through x and finally obtaining x_0. In other words, if we can make an accurate estimate of $q(x_{t-1}|x_t)$ at each step, then we can start with a random noise $x_T \sim \mathcal{N}(0,\mathbf{I})$, and gradually denoise it to generate a target sample. The reverse is the process of data generation. Researchers have designed a model p_θ to approximate these conditional probabilities in order to run the reverse diffusion process [51].

Reverse_Process Approximation

$$p_\theta(x_{0:T}) = p(x_t)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t). \tag{8}$$

Similar to the forward process, each step of the reverse process is based on the previous step. The difference is that the mean $\mu_\theta(x_t,t)$ and variance $\mathbf{\Sigma}_\theta(x_t,t)$ of each step in the inverse process are obtained by model inference. The detailed hyperparameter settings for the diffusion model are described in the experimental section.

Conditional information is very important in time series imputation. In our work, we use the classifier-free scheme to add conditional information to the diffusion and reverse processes, the symbol c represents the implicit information that we extract. It can be represented as follows:

$$p_\theta(x_{t-1}|x_t) \xrightarrow{\text{add}} p_\theta(x_{t-1}|x_t,c). \tag{9}$$

In contrast to the classifier-guidance scheme, in the classifier-free scheme we use, instead of retraining a classifier to provide the classification gradient, it is directly define as

$$p(x_{t-1}|x_t,c) = \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t,c) \tag{10}$$

$$p_\theta(x_{t-1}|x_t,c) = \mathcal{N}(x_{t-1};\mu(x_t,t,c),\sigma^2\mathbf{I}). \tag{11}$$

The mean value $\mu_\theta(x_t,t,c)$ is a model parameterized by θ, and the variance σ^2 is a constant. To generate a target sample with conditional information c, we first sample x_T from the standard normal distribution, then draw $x_{t-1} \sim p_\theta(x_{t-1}|x_t,c)$ for $t = T, T-1,\ldots,1$, and finally inference target sample x_0. The mean value $\mu_\theta(x_t,t,c)$ is a model parameterized by θ, and the variance σ^2 is a constant. To generate a target sample x_0 with conditional information c, we first sample x_T from the standard normal distribution $\mathcal{N}(0,1)$. Then, in the reverse diffusion process, we draw $x_{t-1} \sim p_\theta(x_{t-1}|x_t,c)$ for $t = T, T-1,\ldots,1$, where the conditional probability $p_\theta(\cdot|x_t,c)$ is parameterized by θ and conditioned on c. Finally, we obtain the target sample x_0 after the sequential reverse sampling steps.

D. Multiscale Implicit Feature

background

Most works take observed data as conditional information to optimize model predictions in time series imputation tasks. Time series exhibit cyclical variations that can be used to better analyze temporal characteristics. They have obvious trend features and significant periodicity in both the short term and long term, which is the basis for data imputation. Periodicity in time series is very complex. Daily, weekly, and annual cycles may be intertwined and correlated. Taking the PeMS traffic dataset as an example, the dataset presents short-term trends and long-term cyclical characteristics. As shown in Fig. 3(a), the traffic speed data here is analyzed at the granularity of days and show a weekly correlation in these 60 days of traffic speed data. Due to the different working patterns, the trends of traffic speeds on weekdays and weekends differ significantly. Fig. 3(b) displays the weekly traffic speed patterns, which reveal a clear daily periodic trend. Fig. 3(c) illustrates the short-term (daily) traffic variations that reveal a clear peak-to-valley trend corresponding to human activities. For instance, during the period from 16:00 to 18:00, the traffic speed reaches its lowest point when the increased travel demand of people for work, leisure activities, and other purposes, results a large number of vehicles to appear on the road, even cause congestion. During the night hours from 00:00 to 05:00, the speed of vehicles is maintained at a relatively high level because people are less active and there are not many vehicles on the road.

Long_term
Short_term

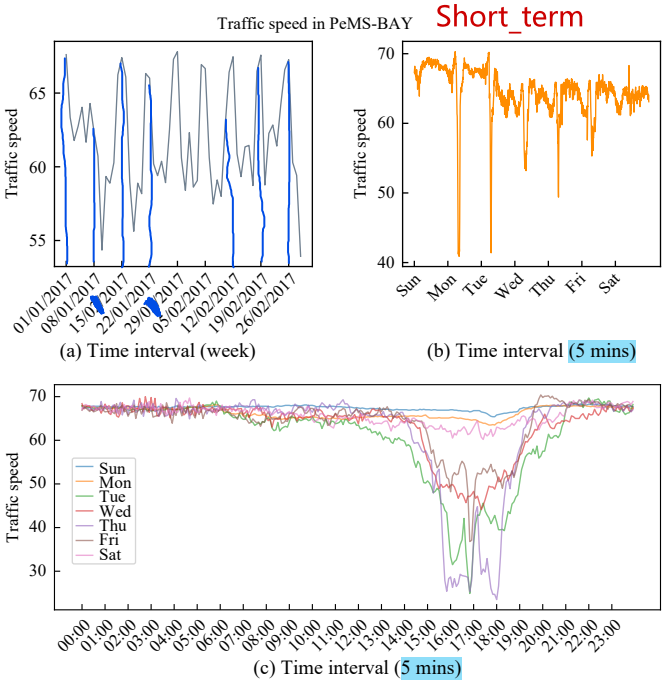


Fig. 3. Short-term trends and long-term cyclical characteristics across different granularities: Weekly, daily, and hourly intervals in three subplots.

Analyses of traffic data

The value of k features at time t is denoted by $x_t^d \in \mathbb{R}$. $f^k = (x_1^k, x_2^k, x_3^k, \ldots, x_t^k) \in \mathbb{R}$ denotes the time series of k features. We extract the implicit feature for each data x_t^k at different time scales and set different granularity factors g to extract implicit feature in different ranges. For instance, the temporal granularity g is set to 1 day for extracting the day of year feature, and set to 1 hour for extracting the hour of day feature. The detail are as follows:

$$x_t^k = \begin{cases} x_{\text{hour}}, & g = 1 \text{ hour} \\ x_{\text{day}}, & g = 1 \text{ day} \\ \ldots & \ldots \\ x_{\text{month}}, & g = 1 \text{ month}. \end{cases} \tag{12}$$

The implicit feature of the original time series is captured. Thus, the model inputs include global information such as

hierarchical time stamps (week, month, and year) as well as agnostic information (holidays, events) and the original time series. The process of implicit feature extraction is shown in Fig. 4.

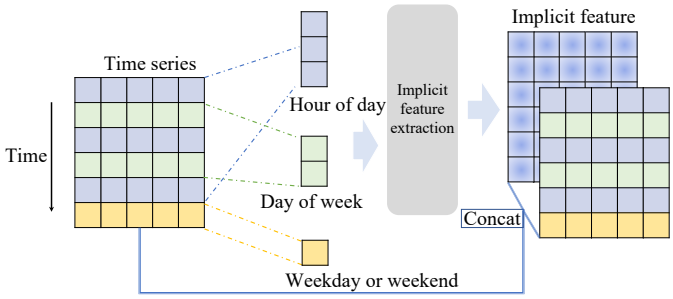


Fig. 4. Implicit information integration: Extracting and fusing features at different granularities.

E. Multiscale Implicit Feature Extraction Module

significance

Effective feature extraction is essential for accurate time series imputation. Time series data are usually collected at fixed intervals, such as 5 minutes, 10 minutes, or 1 hour. However, only a limited number of features can be extracted at a single temporal granularity, and noise during the extraction process can lead to high data density. To solve this problem, it is necessary to extract various traffic trends at multiple time granularities. In this study, we propose a module for implicit extraction based on 1-dimensional fully convolutional network and dilated causal convolution from TCN. The proposed module is designed to overcome the limitations of traditional feature extraction methods and effectively capture implicit dependencies at multi scales hidden in the raw data.

The implicit extraction module incorporates dilated causal convolution to capture local and global temporal relationships. Causal convolution limits the utilization of information from the previous moment to estimate the current moment by manipulating convolutional neurons. This prevents future leakage and makes it suitable for modeling temporal features. To capture multiscale implicit dependencies at different temporal granularities [57], we set expansion factors for the causal convolution to obtain different sizes of perceptual fields. More specifically, given a 1-dimensional time series $x \in \mathbb{R}^D$ and a filter f, the computational procedure of the inflated causal convolution is defined as follows:

$$F(x) = x_t * f(\cdot) = \sum_{i=1}^{n} f(s) \cdot x_{t-d \cdot s} \tag{13}$$

where "$*$" represents the dilated convolution operation, x is the input at time t, d is the dilation factor, and s denotes the number of the layer in which the convolution is located. Generally, the higher the number of layers, the larger the factor is set.

To model the multiscale dependency lying in the raw data, we extract the temporal features at three different granularities, including the minutes of the hour, the hours of the day, and the days of the month. Correspondingly, three implicit representations are learned leveraging the features, denoted as $\mathbf{X}_m^*$, $\mathbf{X}_h^*$, and $\mathbf{X}_d^*$.

$$\mathbf{X}_{\text{implicit}} = concat(\mathbf{X}_m^*, \mathbf{X}_h^*, \mathbf{X}_d^*). \tag{14}$$

Once the implicit features $\mathbf{X}_{\text{implicit}}$ are concatenated, they are passed as input to the three stacked causal dilated convolutions in the proposed implicit extraction module. To ensure that the input and output dimensions of the module are consistent, we incorporate a clip module into in a stacked diffusion causal convolution. Moreover, we use residual connections to maintain the stability of the gradient and avoid the problem of gradient disappearance that may occur when too many layers of causal convolution are stacked.

ResNet

$$output = ReLU(\mathbf{X}_{\text{implicit}} + F(\mathbf{X}_{\text{implicit}})). \tag{15}$$

F. Explicit Extraction Module

In this work, we leverage the structured state space sequence model (S4) to capture long-range temporal dependencies as explicit feature from the original time series. S4 is based on the state space model (SSM), which is defined by the following equation:

$$\begin{aligned} x'(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t). \end{aligned} \tag{16}$$

This equation maps a 1-Dimensional series $u(t)$ to a 1-dimensional output feature $y(t)$ via an N-dimensional latent state $x(t)$. Here, $x'(t)$ represents the time derivative of the state $x(t)$, $u(t)$ is the system input, and $y(t)$ is the system output. Where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ are the state transition matrix, and it is learned by gradient descent optimization. For a given step size Δ, these continuous-time parameters can be converted to discrete-time parameters $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, $\bar{\mathbf{C}}$ and $\bar{\mathbf{D}}$. The output $y(t)$ was calculated by a convolution computation $\bar{\mathbf{K}}$ as $y(t) = \bar{\mathbf{K}} * u$, and $\bar{\mathbf{K}}$ is the SSM kernel filter, which define as: $\bar{\mathbf{K}} = (\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$. It is important to note that S4 restricts the state matrix $\mathbf{A}$ as a normal plus low-rank matrix [58], which allows for faster computation. Furthermore, this parameterization includes the HiPPO matrices [58], which theoretically and empirically allow the SSM to capture long-range dependencies.

The S4 layer contains a bank of SSM with N-dimensional states. We build on the design of the S4 layer and introduce an original feature capture module, the explicit extraction module. The basic structure of our explicit extraction module is shown in Fig. 5.

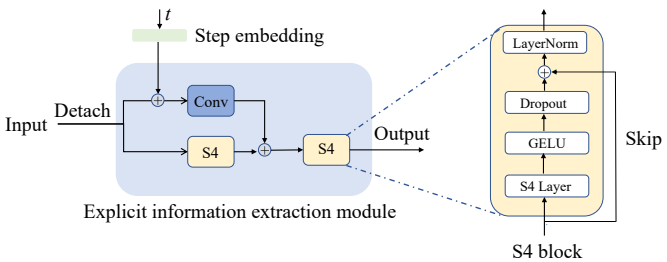


Fig. 5. Explicit extraction module.

IV. Experiments

In this section, we validate the implicit-explicit diffusion

model on three publicly available real time series datasets and compare it with several baseline models. First, we introduce the datasets used in our experiments, which are commonly employed in the traffic field. Next, we select several traffic imputation approaches as the baseline from the field of statistical-based and machine learning. In addition, we incorporate evaluation metrics into our experiments. Finally, we conducted experiments on different datasets with different missing rates.

A. Description of the Traffic Time Series Dataset

1) PeMS-BAY [59]: The dataset originates from the California Transportation Performance Measurements Systems, encompassing a road network comprising 325 traffic sensors situated in the Bay area. It comprises an extensive compilation of traffic data captured at regular five-minute intervals spanning a six-month duration, specifically encompassing the period from January 1 to May 31, 2017. This dataset comprises 52 116 time steps and encompasses 365 nodes. In this study, we partition the initial 50 000 time steps into 250 separate batches, each containing 200 consecutive time steps. To ensure the integrity of the testing process and mitigate the potential impact of information leakage, we directly employ the last 10 batches as the dedicated test set, thereby ensuring equitable and unbiased assessment.

2) METR-LA [59]: The dataset utilized in this study was collected from loop detectors that were installed on county highways in Los Angeles, as referenced by [60]. For the purpose of experimentation, a subset of 207 sensors was selected, and data spanning a four-month period, specifically from March 1 to June 30, 2012, was gathered.

To align with the experimental settings described in [41], a total of 170 samples, each consisting of 200 consecutive time steps, were obtained. This was achieved by selecting the first 34 000 time steps and the first 200 nodes from the original dataset. The training set comprised the initial 150 samples, while the remaining 20 samples were designated as the testing set, following a procedure consistent with the methodology employed for the PeMS-BAY dataset.

The details of the aforementioned datasets are presented in Table II. It should be noted that the final row of the table indicates whether or not there were any missing entries in the dataset at the time of acquisition.

TABLE II
DETAILS OF DATASETS IN EXPERIMENTS

	PeMS-BAY [59]	METR-LA [59]
AREA	Bay Area	Los Angeles
BEGIN DATE	2017/01/01	2012/03/01
END DATE	2017/05/31	2012/06/30
INTERVAL	5 Min	5 Min
TIME STEPS	52 116	34 272
NODES	325	207
MISSING DATA	YES	YES

B. Baseline Methods

1) Mean [2]: Filling the missing data using the current average of other nodes in the area.

2) KNN [11]: In KNN imputation, the k nodes with the highest weights are selected according to the weight ordering in the node adjacency matrix W, and then the average of these nodes is used to impute.

3) MICE [61]: Multiple imputations by chained equations, also known as completely conditional specification and sequential regression multivariate imputation, are a useful method for producing imputations based on a collection of imputation models, one for each variable having missing values.

4) VAR [62]: The vector autoregressive (VAR) model is a multivariate time series model that links the most recent observations of one variable to its historical data and historical data for other variables in the system.

5) rGAIN [63]: It could be viewed as an imputer with recurrent encoders that operate in both directions.

6) BRITS [35]: A novel method to use recurrent dynamics to impute the missing values in multivariate time series effectively.

7) MPGRU [64]: The message-pass GRU, is a one-step-ahead GNN-based predictor similar to DCRNN [59].

8) SSSD [41]: A diffusion-based model, which incorporates structured state space models to capture dependencies of long-term time series.

9) Median: Filling the missing data using the median value from the total training dataset.

10) MF [12]: Matrix factorization decomposes a matrix $\mathbf{X}$ with missing values into two (or more) matrices, and then multiplies these decomposed matrices to obtain an approximation $\mathbf{X}'$ of the original matrix, which we use to fill the missing part of the original matrix $\mathbf{X}$.

11) SAITS [38]: A novel self-attention-based model to impute missing values in multivariate time series.

C. Evaluation Metric

In this section, several widely used metrics were selected to evaluate the performance of our imputation models.

$$MAE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}|\hat{y}_{ij} - y_{ij}|}{\sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}} \tag{17}$$

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}(\hat{y}_{ij} - y_{ij})^2}{\sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}} \tag{18}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}(\hat{y}_{ij} - y_{ij})^2}{\sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}}} \tag{19}$$

$$MRE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}\frac{|\hat{y}_{ij} - y_{ij}|}{\hat{y}_{ij}}}{\sum_{i=1}^{M}\sum_{j=1}^{N} m_{ij}} \tag{20}$$

where M is the total number of test set, N is the dimension of each test set, $\hat{y}_{ij}$ and y_{ij} are the imputed data and the real data, respectively. The m indicates whether the data is missing or not.

D. Time Series Imputation Experiments

We implement the implicit-explicit diffusion model using PyTorch, and train it with the Adam optimizer. All experiments are conducted in the Intel(R) Xeon(R) CPU and two 32 GB Tesla V100 GPUs. In order to speed up the experiment process, we run the experiments with different missing rates on each GPU.

Experiments on PeMS-BAY and METR-LA: The PeMS-BAY dataset is a popular dataset in the field of traffic data imputation and prediction. Since the dataset itself has missingness, it is possible to choose the missing position to fill, and this leads to swings in experimental results. Each experiment was run five times to overcome this problem, and the results were averaged. We utilize the first 50 000 time steps of the PeMS-BAY dataset in this experiment. Moreover, we use the last 2000 time steps as the test set and the rest as the training set. Before feeding the model, we normalize the dataset and divide it into 200 time steps per 75 nodes as a sample. We use the random missing method to generate a 25% missing pattern, which means that each sample contains 50 missing points to be imputed. Similar to [41], we train our model in about 400 000 iterations, results of the experiment on PeMS-BAY are shown in Table III. All baseline results were collected from [41]. Compared with the SSSD model, our model decreased the MAE from 1.05 to 0.89, which is about 15.24% improvement. For MSE and MRE, the improvement is 38.26% and 29.44%, respectively. It should be noted that we compute the metrics after inverse normalization. Our model takes into account implicit feature at different time granularities, extracts implicit features from them, and fuses them with the original input features. Therefore, it presents competitive performance compared to other methods.

improvements

TABLE III
MODEL EVALUATED ON PEMS-BAY [59] DATASET, MISSING RATE SET TO 25%, THE BEST RESULTS ARE IN BOLD, († DENOTES THE REPRODUCED RESULTS)

Dataset	Model	MAE	MSE	MRE (%)
PeMS-BAY	Mean	5.42 ± 0.00	86.59 ± 0.00	8.67 ± 0.00
	KNN	4.30 ± 0.00	49.8 ± 0.00	8.67 ± 0.00
	MF	3.29 ± 0.01	51.39 ± 0.64	5.27 ± 0.02
	MICE	3.09 ± 0.02	31.43 ± 0.01	4.95 ± 0.01
	VAR	1.30 ± 0.00	6.52 ± 0.01	2.07 ± 0.01
	rGAIN	1.88 ± 0.02	10.37 ± 0.20	3.01 ± 0.04
	BRITS	1.47 ± 0.00	7.94 ± 0.03	2.36 ± 0.00
	MPGRU	1.11 ± 0.00	7.59 ± 0.02	1.77 ± 0.00
	SSSD†	1.05 ± 0.02	3.45 ± 0.00	2.48 ± 0.01
	Ours	**0.89** ± 0.00	**2.13** ± 0.02	**1.75** ± 0.00

The visualization of the comparison results between our model and SSSD [41] is shown in Fig. 6. In the upper part of the figure, the "∗" represents the ground truth. The SSSD model and our model are represented by the blue and pink dashed lines respectively. For the PeMS-BAY dataset, we present imputation instances. The confidence bands derived from our model with the quantiles set at 0.95 are shown in lighter blue. The lighter pink represents the result of SSSD imputation with the same quantiles. Our model produces accurate imputations with high confidence compared with SSSD. In particular, our model is able to fit the real traffic data well when there are some missing points with large fluctuations.

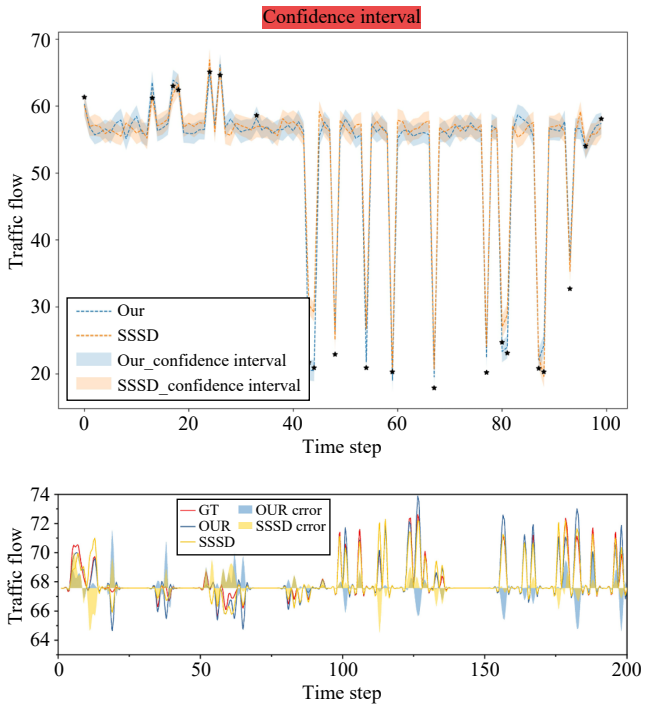


Fig. 6. Comparison results between our model and SSSD [41] visualization.

At the bottom of Fig. 6, we constructed error bars to compare the imputation performance. The ground truth is shown in red, and the blue and yellow represent the imputation performance of our model and SSSD [41], respectively. It can be seen that, for the most part, the data generated by our model is stable with little dispersion and high confidence. The process of splitting the METR-LA dataset is described in the first part of this chapter. Again, we first normalize the dataset, just as in [41]. The data is not inverse normalized during the inference process, from Table IV, it can be seen that our Implicit-Explicit Diffusion Model achieves excellent performance on MAE and MRE. The results illustrate that our model has excellent performance on different time series datasets of traffic.

E. Evaluate Implicit-Explicit Model at Different Missing Rates

In order to further evaluate the performance of our model in the presence of varying levels of missingness beyond imputation of data from a single domain, we conducted experiments on the UCI power dataset with missing rates ranging from 20% to 80%. Unlike PeMS-BAY and METR-LA, it does not contain missing data. This dataset was collected from January 1, 2011 to December 31, 2014. We apply the same division of datasets as in [41]. In this experiment, we trained the implicit-explicit diffusion model with about 200 000 iterations and set the learning rate to 2E–4. Table V summarizes the results of

good performance on different datasets

reasons why we outerperform

TABLE IV
MODEL EVALUATED ON METR-LA [59] DATASET, MISSING RATE SET TO 25%, THE BEST RESULTS ARE IN BOLD, († DENOTES THE REPRODUCED RESULTS)

Dataset	Model	MAE	MSE	MRE (%)
METR-LA	Mean	7.56 ± 0.00	144.22 ± 0.00	13.10 ± 0.00
	KNN	7.88 ± 0.00	129.29 ± 0.00	13.65 ± 0.00
	MF	5.56 ± 0.03	113.46 ± 1.08	9.62 ± 0.05
	MICE	4.42 ± 0.07	55.07 ± 1.46	7.65 ± 0.12
	VAR	2.69 ± 0.00	21.10 ± 0.02	4.66 ± 0.00
	rGAIN	2.83 ± 0.01	20.03 ± 0.09	4.91 ± 0.01
	BRITS	2.34 ± 0.00	**16.46** ± 0.05	4.05 ± 0.00
	MPGRU	2.44 ± 0.00	22.17 ± 0.03	4.22 ± 0.00
	SSSD†	3.04 ± 0.02	26.48 ± 0.07	5.98 ± 0.03
	Ours	**2.27** ± 0.02	22.94 ± 0.03	**3.85** ± 0.00

this experiment, and shows that the implicit-explicit diffusion model achieves the lowest MAE and MSE for all missing rates. As shown in this table, the filling performance of almost all models becomes worse as the missing rate increases. This is because the conditional information gradually decreases. So, we can see how important the the conditional information is to the filling process and that the implicit feature we extracted from the original input strongly complements the conditional information.

F. Ablation Study

In this part, three ablation experiments are leveraged to discuss the rationality of our model architecture. Table VI contains the results we obtained to find the impacts of the implicit extraction model and the multi-granularity implicit feature in our model. The differences between each set of experiments are described as follows: 1) We take SSSD [41] as the benchmark model, which has no implicit extraction module. The experiments were conducted on PeMS-Bay datasets, and the missing rates were set to 25%; 2) A model of this setting that implements implicit feature extraction with single granularity implicit feature based on setting A; 3) To quantify the contribution of the implicit feature, based on setting B, we added implicit feature with multiple granularities. As shown in Table VI, the performance of the model gradually improves with the addition of the implicit extraction module and the increase the feature dimension. It is evident that the performance of the proposed model improves when the implicit extraction module is incorporated and the implicit feature dimension increases.

G. Experiment Results Analysis

From the experimental results, it can be observed that our proposed model demonstrates superior performance. Next, we analyze the possible reasons for these results from a theoretical perspective.

In our imputation task, the time series is denoted as x and the implicit information as c, then $q(x,c)$ represents their joint distribution. Meanwhile, $p_{\theta,E}(x)$ denotes a model dominated by learnable parameters θ and conditional information E. During unconditional model training, $p_{\theta,E}(x)$ directly approximates the marginal data distribution $q(x)$, where E is a shared encoding representation for all data. Given an upper bound on dissimilarity D, the objective during unconditional model training is to optimize

$$\arg\min_{\theta,E} \mathbb{E}_{x\sim q(x)}\left[D(q(x)\|\|p_{\theta,E}(x))\right]. \tag{21}$$

When incorporating implicit information for learning, the embedded encoding E is used to represent the encoding of the input implicit information. Our implicit information capture module $\phi \in \Phi$ is employed to encode implicit information. $p_{\theta,E}(x|E=\phi(c))$ denotes the model input probability regarding x given the condition. During the training process, we aim to approximate the data distribution x under the given condition c by fitting the main model parameters θ and the implicit feature extraction module ϕ. The objective during conditional model training is to optimize:

$$\arg\min_{\theta,\phi} \mathbb{E}_{x\sim q(x),c}\left[D(q(x|c)\|p_{\theta,E}(x|E=\phi(c)))\right]. \tag{22}$$

We compare the expressive capabilities of the model under the same parameters before and after incorporating implicit information. The conditional generative model generates samples from $p_{\theta,\phi}(x)=\mathbb{E}_{q(c)}[p_{\theta,\phi}(x|c)]=\mathbb{E}_{q(c)}[p_{\theta,E}(x|E=\phi(c))]$. Therefore, $p_{\theta,\phi}(x)$ can be viewed as a mixture of several unconditional models. Although the model shares the same backbone network $p_{\theta,E}(x)$ at this point, the conditional generative model exhibits stronger expressive capabilities compared to the unconditional generative model.

For the computation cost, we have calculated the parameter size of the proposed model and tested its time consumption during training, comparing it with different training datasets. As shown in Table VII, we acknowledge the limitation of our proposed model in terms of long training and testing times, as well as the relatively large model size. Therefore, exploring methods for feature compression and efficient sampling will be one of the key focuses of our future work.

V. CONCLUSION

This paper proposes a new method of imputing missing traffic data with implicit feature based on a diffusion model. Implicit extraction module is designed to capture implicit dependencies at diverse temporal granularities hidden in raw traffic data, which is beneficial to impute missing data more accurately. Extensive experiments are conducted on traffic time series and electricity consumption datasets, and the experimental results show that the proposed implicit-explicit diffusion model captures the implicit dependencies of diverse temporal granularities used to impute time series data accurately.

Pro

The training and inference process of our proposed method is comparatively slow. As future works, first, we plan to design another diffusion-based model that has faster sampling and training speeds. Second, we plan to introduce closed-loop imputation-prediction: Which combines the traffic imputation and prediction processes and provides model performance feedback from the prediction results.

Con/Still need improving

TABLE V
PERFORMANCE COMPARISON BETWEEN METHODS ON THE UCI ELECTRICITY DATASET ACROSS DIFFERENT MISSING RATES FROM 20% TO 80%. **MAE** AND **RMSE** ARE REPORTED AT DIFFERENT MISSING RATES. VALUES IN BOLD ARE THE BEST. ALL BASELINE RESULTS WERE COLLECTED FROM [38]

Dataset	Model	20% missing		30% missing		40% missing		50% missing		60% missing		70% missing		80% missing	
UCI ELECTRICITY	Median	2.053	2.726	2.055	2.732	2.058	2.734	2.053	2.728	2.057	2.734	2.050	2.726	2.059	2.734
	Last	1.012	1.547	1.018	1.559	1.025	1.578	1.032	1.595	1.040	1.615	1.049	1.640	1.059	1.663
	M-RNN	1.242	1.854	1.258	1.876	1.269	1.884	1.283	1.902	1.298	1.912	1.305	1.928	1.318	1.951
	GP-VAE	1.124	1.502	1.057	1.571	1.090	1.578	1.097	1.572	1.101	1.616	1.037	1.598	1.062	1.621
	BRITS	0.928	1.395	0.943	1.435	0.996	1.504	1.037	1.538	1.101	1.602	1.090	1.617	1.138	1.665
	Transformer	0.843	1.318	0.846	1.321	0.876	1.387	0.859	1.410	0.891	1.404	0.920	1.437	0.924	1.472
	SAITS	0.763	1.187	0.790	1.223	0.869	1.314	0.876	1.377	0.892	1.328	0.898	1.273	0.908	1.327
	Ours	**0.272**	**0.389**	**0.297**	**0.424**	**0.334**	**0.477**	**0.378**	**0.540**	**0.450**	**0.655**	**0.541**	**0.776**	**0.732**	**1.049**

TABLE VI
ABLATION EXPERIMENTS ON PEMS-BAY [59], MISSING RATE SET TO 25%, THE BEST RESULTS ARE IN BOLD

Setting	MAE	MSE	MRE (%)
A	1.053	3.425	2.642
B	0.969	2.708	2.074
C	**0.891**	**2.133**	**1.752**

TABLE VII
THE COMPUTATION COST ON PEMS-BAY AND METR-LA

	#Parameter	Training time (s /100 iter)	Inference time
PeMS-BAY	1.5 G	00:01:05	00:02:41
METR-LA	605 MB	00:02:34	00:02:37

REFERENCES

[1] D. Xu, H. Peng, C. Wei, X. Shang, and H. Li, "Traffic state data imputation: An efficient generating method based on the graph aggregator," *IEEE Trans. Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13084–13093, 2022.

[2] M. Kantardzic, "Data mining: Concepts, models, methods, and algorithms," *Technometrics*, vol. 45, no. 3, p. 277, 2003.

[3] Q. Miao, Y. Lv, M. Huang, X. Wang, and F.-Y. Wang, "Parallel learning: Overview and perspective for computational learning across Syn2Real and Sim2Real," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 3, pp. 603–631, 2023.

[4] W. Zou, Y. Sun, Y. Zhou, Q. Lu, Y. Nie, T. Sun, and L. Peng, "Limited sensing and deep data mining: A new exploration of developing city-wide parking guidance systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 1, pp. 198–215, 2022.

[5] C. Zhao, Y. Lv, J. Jin, Y. Tian, J. Wang, and F.-Y. Wang, "Decast in transverse for parallel intelligent transportation systems and smart cities: Three decades and beyond," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 6–17, 2022.

[6] S. Feng, Z. Song, Z. Li, Y. Zhang, and L. Li, "Robust platoon control in mixed traffic flow based on tube model predictive control," *IEEE Trans. Intelligent Vehicles*, vol. 6, no. 4, pp. 711–722, 2021.

[7] X. Ge, Q.-L. Han, X.-M. Zhang, and D. Ding, "Communication resource-efficient vehicle platooning control with various spacing policies," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 2, pp. 362–376, 2023.

[8] J. Zhan, Z. Ma, and L. Zhang, "Data-driven modeling and distributed predictive control of mixed vehicle platoons," *IEEE Trans. Intelligent Vehicles*, vol. 8, no. 1, pp. 572–582, 2023.

[9] J. Wang, J. Wang, and Q.-L. Han, "Receding-horizon trajectory planning for under-actuated autonomous vehicles based on collaborative neurodynamic optimization," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 11, pp. 1909–1923, 2022.

[10] V. G. Stepanyants and A. Y. Romanov, "A survey of integrated simulation environments for connected automated vehicles: Requirements, tools, and architecture," *IEEE Intelligent Transportation Systems Magazine*, vol. 16, no. 2, pp. 6–22, 2023.

[11] A. T. Hudak, N. L. Crookston, J. S. Evans, D. E. Hall, and M. J. Falkowski, "Nearest neighbor imputation of species-level, plot-scale forest structure attributes from lidar data," *Remote Sensing of Environment*, vol. 112, no. 5, pp. 2232–2245, 2008.

[12] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[13] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2012.

[14] X. Chen, M. Lei, N. Saunier, and L. Sun, "Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation," *IEEE Trans. Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12301–12310, 2021.

[15] X. Chen, Z. He, and J. Wang, "Spatial-temporal traffic speed patterns discovery and incomplete data recovery via SVD-combined tensor decomposition," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 59–77, 2018.

[16] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[17] S. Harford, F. Karim, and H. Darabi, "Generating adversarial samples on multivariate time series using variational autoencoders," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 9, pp. 1523–1538, 2021.

[18] T. Lintonen and T. Räty, "Self-learning of multivariate time series using perceptually important points," *IEEE/CAA J. Autom. Sinica*, vol. 6, p. 1318, 2019.

[19] J. Liu, F. Zheng, X. Liu, and G. Guo, "Dynamic traffic flow prediction based on long-short term memory framework with feature organization," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 221–236, 2022.

[20] Z. Li, Y. Li, and L. Li, "A comparison of detrending models and multiregime models for traffic flow prediction," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 34–44, 2014.

[21] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: Recent advances and new challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018.

[22] S. Choi, H. Shon, and K. Huh, "Interpretable vehicle speed estimation based on dual attention network for 4WD off-road vehicles," *IEEE Trans. Intelligent Vehicles*, vol. 9, no. 1, pp. 151–164, 2023.

[23] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[24] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 168–181, 2016.

[25] J. Xi, F. Zhu, P. Ye, Y. Lv, H. Tang, and F.-Y. Wang, "HMDRL: Hierarchical mixed deep reinforcement learning to balance vehicle supply and demand," *IEEE Trans. Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21861–21872, 2022.

[26] J. Xi, F. Zhu, Y. Chen, Y. Lv, C. Tan, and F. Wang, "DDRL: A decentralized deep reinforcement learning method for vehicle repositioning," in *Proc. IEEE Int. Intelligent Transportation Systems Conf.*, 2021, pp. 3984–3989.

[27] J. Pan, C. Li, Y. Tang, W. Li, and X. Li, "Energy consumption prediction of a CNC machining process with incomplete data," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 5, pp. 987–1000, 2021.

[28] W. Du, B. Li, J. Chen, Y. Lv, and Y. Li, "A spatiotemporal hybrid model for airspace complexity prediction," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 2, pp. 217–224, 2023.

[29] X. Jiang, X. Kong, and Z. Ge, "Augmented industrial data-driven modeling under the curse of dimensionality," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 6, pp. 1445–1461, 2023.

[30] J. Yoon, W. R. Zame, and M. van der Schaar, "Estimating missing data in temporal data streams using multi-directional recurrent neural networks," *IEEE Trans. Biomedical Engineering*, vol. 66, no. 5, pp. 1477–1490, 2019.

[31] X. Yao, Y. Gao, D. Zhu, E. Manley, J. Wang, and Y. Liu, "Spatial origindestination flow imputation using graph convolutional networks," *IEEE Trans. Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7474–7484, 2021.

[32] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," arXiv preprint arXiv: 1906. 00121, 2019.

[33] Y. Liu, R. Yu, S. Zheng, E. Zhan, and Y. Yue, "NAOMI: Nonautoregressive multiresolution sequence imputation," arXiv preprint arXiv: 1901.10946, 2019.

[34] M. Morsali, E. Frisk, and J. Åslund, "Spatio-temporal planning in multivehicle scenarios for autonomous vehicle using support vector machines," *IEEE Trans. Intelligent Vehicles*, vol. 6, no. 4, pp. 611–621, 2021.

[35] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: Bidirectional recurrent imputation for time series," arXiv preprint arXiv: 1805.10572, 2018.

[36] Q. Suo, W. Zhong, G. Xun, J. Sun, C. Chen, and A. Zhang, "GLIMA: Global and local time series imputation with multi-directional attention learning," in *Proc. IEEE Int. Conf. Big Data*, 2020, pp. 798–807.

[37] Z. Wei, H. Zhao, Z. Li, X. Bu, Y. Chen, X. Zhang, Y. Lv, and F.-Y. Wang, "STGSA: A novel spatial-temporal graph synchronous aggregation model for traffic prediction," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 1, pp. 226–238, 2023.

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

[39] Y. Liu, B. Tian, Y. Lv, L. Li, and F.-Y. Wang, "Point cloud classification using content-based transformer via clustering in feature space," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 1, pp. 1–2, 2023.

[40] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "CSDI: Conditional scorebased diffusion models for probabilistic time series imputation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24804–24816, 2021.

[41] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured state space models," arXiv preprint arXiv: 2208.09399, 2022.

[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communi. the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[43] Y. Luo, X. Cai, Y. Zhang, J. Xu, *et al.*, "Multivariate time series imputation with generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 1603–1614, 2018.

[44] Y. Lv, Y. Chen, L. Li, and F.-Y. Wang, "Generative adversarial networks for parallel transportation systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 3, pp. 4–10, 2018.

[45] K. Sarda, A. Yerudkar, and C. D. Vecchio, "Missing data imputation for real time-series data in a steel industry using generative adversarial networks," in *Proc. Ann. Conf. IEEE Industrial Electronics Society*, 2021, pp. 1–6.

[46] X. Xiao, Y. Zhang, S. Yang, and X. Kong, "Efficient missing counts imputation of a bike-sharing system by generative adversarial network," *IEEE Trans. Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13443–13451, 2022.

[47] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Trans. Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1624–1630, 2020.

[48] J. Chen, K. Wu, Y. Yu, and L. Linbo, "CDP-GAN: Near-infrared and visible image fusion via color distribution preserved GAN," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 9, ppp. 1698–1701, 2022.

[49] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Machine Learning*, 2015, pp. 2256–2265.

[50] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," arXiv preprint arXiv: 1907.05600, 2020.

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

[52] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," arXiv preprint arXiv: 2009.09761, 2020.

[53] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv: 1803.01271, 2018.

[54] G. Guo, W. Yuan, J. Liu, Y. Lv, and W. Liu, "Traffic forecasting via dilated temporal convolution with peak-sensitive loss," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 1, pp. 48–57, 2023.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," arXiv preprint arXiv: 1706.03762, 2023.

[56] K. Liu, Z. Ye, H. Guo, D. Cao, L. Chen, and F.-Y. Wang, "Fiss GAN: A generative adversarial network for foggy image semantic segmentation," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 8, pp. 1428–1439, 2021.

[57] H. Huang, G. Zhou, N. Liang, Q. Zhao, and S. Xie, "Diverse deep matrix factorization with hypergraph regularization for multiview data representation," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 11, pp. 2154–2167, 2022.

[58] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "HIPPO: Recurrent memory with optimal polynomial projections," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1474–1487, 2020.

[59] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," arXiv preprint arXiv: 1707.01926, 2017.

[60] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Communi. the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[61] I. R. White, P. Royston, and A. M. Wood, "Multiple imputation using chained equations: Issues and guidance for practice," *Statistics in Medicine*, vol. 30, no. 4, pp. 377–399, 2011.

[62] L. Ljung, "Prediction error estimation methods," *Circuits, Systems and Signal Processing*, vol. 21, pp. 11–21, 2002.

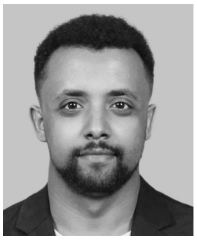
[63] X. Miao, Y. Wu, J. Wang, Y. Gao, X. Mao, and J. Yin, "Generative semi-supervised learning for multivariate time series imputation," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 8983–8991.

[64] K. Swanson, "Message passing neural networks for molecular property prediction," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, USA, 2019.

Bo Lu received the B.E degree in computer science and technology from Chengdu University of Information Technology in 2021, and the M.E. degree in artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, as well as the School of Artificial Intelligence, University of Chinese Academy of Sciences in 2024. He is currently an Algorithm Engineer at Baidu Inc. His research interests include artificial intelligence-generated content and computer vision.

Qinghai Miao (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the Graduate University of Chinese Academy of Sciences in 2007. From 2017 to 2018, he was a Visiting Scholar with the School of Informatics, the University of Edinburgh, UK. He is currently an Associate Professor with the School of Artificial Intelligence, University of Chinese Academy of Sciences. His research interests include parallel intelligence, machine learning, computer vision, intelligent transportation systems.

Yahui Liu received the B.E. degree in electronic information science and technology from Xiamen University in 2021, and the M.E. degree in computer technology from the Institute of Automation, Chinese Academy of Sciences in 2024. She is currently an Algorithm Engineer specializing in autonomous driving at Meituan. Her research interests include motion prediction and vision-based autonomous driving.



Tariku Sinshaw Tamir received the B.Sc. degree in electrical engineering from Haramaya University, Ethiopia in 2011, the M.Sc. degree in control engineering from Addis Ababa University, Ethiopia in 2015, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences in 2023. He is currently a Postdoctoral Researcher at Guangdong University of Technology. His research interests include modeling, control, and optimization of complex systems; 3D printing; and smart manufacturing. He was a recipient of the Excellent International Student Award in 2021 and 2022 from the University of Chinese Academy of Sciences. He has authored more than 20 referred journal and conference papers.

Hongxia Zhao (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the Chinese Academy of Sciences in 2009. She is currently an Assistant Professor with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences. Her research interests include intelligent control, data analysis, and traffic modeling and prediction.

Xiqiao Zhang received the B.E. degree in traffic engineering, the M.E. degree in transportation planning and management, and the Ph.D. degree in bridge and tunnel engineering from Harbin Institute of Technology, in 2001, 2004, and 2008, respectively. He is currently an Associate Professor with the Department of Transportation Engineering, the School of Transportation Science and Technology, Harbin Institute of Technology. His research interests include traffic data analysis, traffic control, dynamic traffic modeling.

Yisheng Lv (Senior Member, IEEE) received the B.E. and M.E. degrees in transportation engineering from Harbin Institute of Technology in 2005 and 2007, respectively, and the Ph.D. degree in control theory and control engineering from the Chinese Academy of Sciences in 2010. He is a Professor with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include traffic data analysis, dynamic traffic modeling, and parallel traffic management and control systems.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, USA in 1990. He joined the University of Arizona, USA in 1990, and became a Professor and the Director of the Robotics and Automation Lab and Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), and in 2002, was appointed as the Director of the Key Laborabory of Complex Systems and Intelligence Science, CAS. From 2006 to 2010, he was Vice President for Research, Education, and Academic Exchanges with the Institute of Automation, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems. His research interests include methods and applications for parallel systems, social computing, parallel intelligence, and knowledge automation. He was the Founding Editor-in-Chief (EiC) of the *International Journal of Intelligent Control and Systems* (1995–2000), Founding EiC of *IEEE ITS Magazine* (2006–2007), EiC of *IEEE Intelligent Systems* (2009–2012), EiC of *IEEE Transactions on ITS* (2009–2016), and EiC of *IEEE Transactions on Computational Social Systems* (2017–2020). He is currently the EiC of *IEEE Transactions on Intelligent Vehicles*, Founding EiC of *IEEE/CAA Journal of Automatica Sinica*, and *Chinese Journal of Command and Control*. Since 1997, he was the General or Program Chair of more than 20 IEEE, INFORMS, ACM, and ASME conferences. He was the President of IEEE ITS Society (2005–2007), Chinese Association for Science and Technology, Beijing, China, in 2005, American Zhu Kezhen Education Foundation (2007–2008), and Vice President of the ACM China Council (2010–2011). Since 2008, he has been the Vice President and Secretary General of Chinese Association of Automation. Dr. Wang has been elected as Fellow of INCOSE, IFAC, ASME, and AAAS. In 2007, he was the recipient of the National Prize in Natural Sciences of China, Outstanding Scientist by ACM for his research contributions in intelligent control and social computing, IEEE ITS Outstanding Application and Research Awards in 2009, 2011and 2015, respectively, and IEEE SMC Norbert Wiener Award in 2014.