

# Spatiotemporal Extrapolation through Conditional Diffusion Probability Models

En Wang, Qinglun Meng, Wenbin Liu, Bo Yang, and Jie Wu, Fellow, IEEE

**Abstract**—In recent years, spatiotemporal data has played a crucial role in weather, transportation, and disease transmission within the context of the Internet of Things (IoT). However, due to cost constraints and sensor failures, many regions lack observational data and remain unmonitored. This poses a challenge to the generalization of the model which is often addressed by spatiotemporal completion methods. However, most existing interpolation and completion methods are limited to the data distribution of the training regions and struggle to generalize to out-of-distribution scenarios. This paper addresses the challenge of generalization, particularly in scenarios that require inference from regions that have never been observed before. To overcome this limitation, we propose an inductive generative model for spatiotemporal extrapolation. Our approach is based on a denoising diffusion probabilistic model, incorporating attention mechanisms guided by non-local features and dynamic topology information. This enables our model to generalize to previously unseen regions. Empirical evaluations of three datasets in real-world and cross-city evaluations demonstrate the superior performance of our approach over state-of-the-art methods.

**Index Terms**—Spatiotemporal Extrapolation, Climate Science, Diffusion Probabilistic Models, Dynamic Graph information, Context, Geology

## I. Introduction

**S**PATIOTEMPORAL data, characterized by inherent spatial and temporal patterns, plays a crucial role in numerous real-world applications within the realm of the Internet of Things (IoT), such as air quality monitoring [1]–[4], traffic status forecasting [5], [6], social network [7]. Traditionally, the collection of spatiotemporal data relies heavily on fixed sensor networks. Although mobile sensing technologies, such as Mobile CrowdSensing [8]–[10], have emerged in recent years, data collection remains limited by both the distribution of fixed sensors and the mobility of sensing participants. Note that spatiotemporal data is often sparse and unevenly distributed across space and time, with no sensor network—fixed or mobile—capable of fully covering all regions. This has led to significant research focused on leveraging spatiotemporal correlations to infer missing data, aiming to fill the gaps in data coverage. Numerous methods have been proposed

En Wang, Qinglun Meng, Wenbin Liu, and Bo Yang are with the College of Computer Science and Technology, Jilin University, Changchun, Jilin, CN (e-mail: wangen@jlu.edu.cn; mengql22@mails.jlu.edu.cn; liuwenbin@jlu.edu.cn; ybo@jlu.edu.cn).

Jie Wu is with the China Telecom Cloud Computing Research Institute, Beijing, 100088, China, and with the Department of Computer and Information Sciences, Temple University, 1925 N. 12th St., Philadelphia, PA 19122, USA (e-mail: jiewu@temple.edu).

Corresponding author: Wenbin Liu.

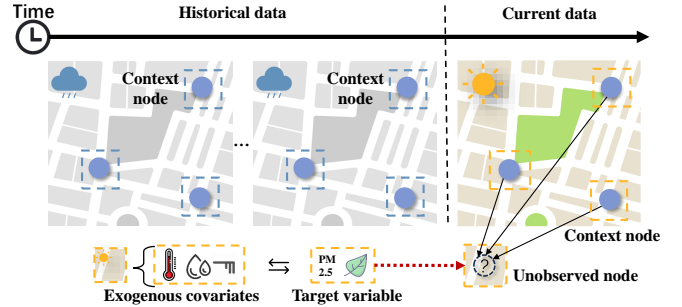


Fig. 1: Spatiotemporal Extrapolation. The diagrams depict the prediction of spatiotemporal target nodes that have never been seen or trained before, leveraging identical spatiotemporal context nodes and corresponding exogenous covariates.

for spatiotemporal data imputation, including approaches based on compressed sensing, matrix factorization [11], and deep learning [12], [13], all of which have achieved promising results. These methods typically operate on a pre-defined grid structure, assuming that data can be collected from a fixed set of regions at each time period. By exploiting spatial correlations between regions and temporal correlations within a single region, they can infer missing spatiotemporal data. However, these methods are largely transductive, meaning they struggle handling newly added or dynamically changing sensing regions. In practice, as shown in Fig. 1, target sensing areas often change over time and may even have not been observed. Therefore, tackling the challenge of inductive spatiotemporal extrapolation in the context of new or dynamically changing target regions is crucial for real-world applications.

When dealing with unseen sensing regions, historical data is often unavailable. Intuitively, generative methods can be employed to synthesize data for such unknown regions. Among these, the diffusion probability model, renowned for its powerful data mining capabilities, has achieved impressive results in sequence data generation [14], [15]. However, most existing spatiotemporal diffusion probability models are transductive during training, relying on data collected from fixed regions where sensors were deployed [6], [13], [14]. In inductive scenarios, due to the need to add and remove noise for all nodes and the complexity of dynamic environments, the training process becomes challenging.

Meanwhile, in spatiotemporal extrapolation tasks, significant non-homogeneity often exists between the target

domain  $D$  and the known domain  $C$  in terms of spatiotemporal distributions, leading to potential inconsistency in the distribution of target variables with the known domain. This distribution shift may arise from differences in temporal distribution, spatial distribution, or feature distribution, requiring models to possess cross-domain transfer capability for accurate inference under distribution inconsistency, presenting another key challenge.

In addition, as new target sensing regions are introduced and existing ones change, the structure of the spatiotemporal graph becomes increasingly dynamic, and the spatial-temporal relationships between sensing data also change [16]. Therefore, how to adapt to these dynamic graph structures and capture the evolving correlations is the third challenge.

To address the aforementioned challenges in applying diffusion probability models to spatiotemporal extrapolation tasks, we propose a Diffusion Probabilistic Model for Spatiotemporal Extrapolation (DSTE). We first introduce graph aggregation to aggregate the information of known regions onto the target region and then perform noise adding training. Through the random sampling training method, we fully train the nodes in the dataset so that the model has the ability of extrapolation. To address the distribution shift across different geographical regions in extrapolation tasks, we then propose a Non-local Factor Learning Module based on the neural process theory. This is a method of using context nodes learning to take the relationship between target variables and covariates as an auxiliary factor to assist inferences. Finally, we design a dynamic graph aggregation module, which uses the relationships between covariates to generate dynamic graphs in real time for auxiliary inferences. In summary, our work makes specific contributions as follows:

- We introduce DSTE, the pioneering diffusion probability model and training method for spatiotemporal extrapolation tasks, enabling data inference in unobserved regions.
- To address the extrapolation challenges, we extract Non-local factors to enhance generalization for unknown regions and integrate them with model training through the loss function.
- To adapt to dynamic topological structures in extrapolation scenarios, we design static subgraph sampling with dynamic topology learning to capture time-varying graph features and strengthen topological adaptation.
- Our model excels in spatiotemporal extrapolation, outperforming others with an impressive reduction in mean absolute error (MAE) on three real-world datasets.

The source code for this study can be accessed at the following repository<sup>1</sup>.

## II. PRELIMINARIES

In this section, we present definitions for the different terms and concepts associated with spatiotemporal data discussed in the article. Following that, we provide a concise introduction to the Denoising Diffusion Probabilistic Model (DDPM).

### A. Definitions and Notations

**Definition 1 (Graph).** We represent a graph as  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  denotes the set of nodes,  $|\mathcal{V}|$  represents the number of vertices in the set, and  $\mathcal{E}$  signifies the set of weighted edges connecting the nodes in the graph.  $A$  represents the adjacency matrix of graph  $\mathcal{G}$ .  $A_s$  specifically designates the static matrix detailing weighted connections influenced by spatial distances among individual nodes within the graph.

**Definition 2 (Spatiotemporal data).** We formalize spatiotemporal data as a sequence  $Y_{1:L} = \{Y_1, Y_2, \dots, Y_L\} \in \mathbb{R}^{K \times L \times dy}$  over consecutive time, where  $Y_l \in \mathbb{R}^{K \times dy}$  is the values observed at time  $l$  by  $K$  observation nodes, such as air monitoring stations and traffic sensors. Here,  $dy$  represents the feature dimension of  $Y_{1:L}$ .

**Definition 3 (Exogenous covariates).** Exogenous covariates are denoted as a sequence  $X_{1:L} = \{X_1, X_2, \dots, X_L\} \in \mathbb{R}^{K \times L \times dx}$ , where  $dx$  is the number of channels. They contribute to the learning process as they exhibit significant correlations with node data. These exogenous covariates are often readily available from diverse sources, such as weather stations providing data on weather conditions, which can influence air pollutant data. For all spatiotemporal data within the set of nodes  $X_i \in \mathbb{R}^K$ , we utilize the pairs  $\{Y_{i,1:L}, X_{i,1:L}\}$ . Here,  $Y_{i,1:L}$  represents the target values that we aim to infer for the corresponding nodes, while  $X_{i,1:L}$  denotes the variables that are more easily accessible and are related to  $Y_{i,1:L}$ .

**Definition 4 (Context and target sets).** In our problem, the set of sites can be categorized into two types. One is the set of context nodes, denoted as  $\mathcal{C}$ , where  $\mathcal{C} = \{Y_{i,1:L}, X_{i,1:L}\}_{i=1}^N$ . Within set  $\mathcal{C}$ , all the data  $\{Y_{i,1:L}, X_{i,1:L}\}$  of the nodes are known. The other type is the set of target nodes, denoted as  $\mathcal{D} = \{Y_{i,1:L}, X_{i,1:L}\}_{i=1}^M$ , representing the target values and exogenous covariates of the regions we aim to infer.  $N$  and  $M$  respectively correspond to the node quantities in sets  $\mathcal{C}$  and  $\mathcal{D}$ . In set  $\mathcal{D}$ , only the covariates are known, while the target values are the objectives of our inference.

### B. Diffusion probabilistic models

Diffusion Probabilistic Model, as a generative model, has spawned various theoretical variants, including NCSN [17], DDPM [18], and SGM [19] that involve a training method that incorporates the addition of noise and denoising. This process entails constructing two parameterized Markov chains to diffuse the data with predefined noise and subsequently reconstructing the desired samples from the introduced noise. In the forward process, DDPM gradually distorts the raw data distribution  $x_0 \sim q(x_0)$  to

<sup>1</sup>Code available at: <https://github.com/loiter74/DSTE>

TABLE I: KEY MATHEMATICAL NOTATIONS OF THIS ARTICLE

| Notation      | Description   |
|---------------|---|
| $\mathcal{G}$ | Spatiotemporal graph  |
| $\mathcal{V}$ | Set of nodes in spatiotemporal graph                          |
| $\mathcal{E}$ | Set of edges in spatiotemporal graph                          |
| $\mathcal{D}$ | Target nodes set  |
| $\mathcal{C}$ | Context nodes set   |
| $Y^0$         | Target variable of target nodes                               |
| $Y^t$         | Target variable of target nodes in diffusion step $t$         |
| $X$           | Exogenous covariates of target nodes                          |
| $Y_C$         | Target variable of context nodes                              |
| $X_C$         | Exogenous covariates of context nodes                         |
| $M$           | Number of target nodes  |
| $N$           | Number of context nodes                                       |
| $L$           | Time window in dataset  |
| $A_S$         | Static adjacency graph  |
| $A_{Dyn}$     | Learnable dynamic adjacency graph                             |
| $\mathcal{K}$ | Convolution kernel  |
| $\alpha^t$    | Coefficient for noise addition at diffusion step $t$          |
| $\beta^t$     | Ratio of noise added at step $t$ calculated as $1 - \alpha^t$ |
| $T$           | Total diffusion steps in diffusion process                    |
| $t$           | Diffusion step in diffusion process                           |
| $\epsilon^t$  | Noise in diffusion step $t$                                   |
| $\theta$      | Model parameters  |
| $q$           | Forward process in diffusion                                  |
| $\gamma$      | Attention module gate parameter                               |
| $Z$           | Distribution representation of target nodes                   |
| $Z_C$         | Distribution representation of context nodes                  |
| $\mu$         | The mean of the normal distribution                           |
| $\sigma$      | The variance of the normal distribution                       |

converge to the standard Gaussian distribution  $z_t$  under a pre-designed mechanism. Meanwhile, the reverse chain aims to train a parameterized Gaussian transition kernel to recover the unperturbed data distribution. Mathematically, the definition of the forward process  $q$  is as follows: here we mainly follow the theoretical model proposed by DDPM. The diffusion time steps are denoted with superscripts to avoid confusion between diffusion time steps and temporal time in space-time, such as  $x^t$ .

$$\begin{aligned} q(x^t|x^{t-1}) &= \mathcal{N}(x^t; \sqrt{1 - \beta^t}x^{t-1}, \beta^t I), \\ q(x^{1:T}|x^0) &= \prod_{t=1}^T q(x^t|x^{t-1}), \end{aligned} \quad (1)$$

where  $\beta_t$  is a small constant hyperparameter that controls the variance of the added noise. The  $x^t$  is sampled by  $x^t = \sqrt{\bar{\alpha}^t}x^0 + \sqrt{1 - \bar{\alpha}^t}\epsilon$ , where  $\alpha^t = 1 - \beta^t$ ,  $\bar{\alpha}^t = \prod_{i=1}^t \alpha^i$ , and  $\epsilon$  is the sampled standard Gaussian noise. When  $T$  is large enough,  $q(x^T|x^0)$  is close to a standard normal distribution. The reverse process can be formalized as:

$$\begin{aligned} p_\theta(x^{0:T}|x^T) &= \prod_{t=1}^T p_\theta(x^{t-1}|x^t), \\ p_\theta(x^{t-1}|x^t) &= \mathcal{N}(x^{t-1}; \mu_\theta(x^t, t), \sigma^2 I). \end{aligned} \quad (2)$$

DDPM introduces an effective parameterization of  $\mu_\theta$  and  $\sigma_\theta^2$ . In this work, they can be defined as:

$$\begin{aligned} \mu_\theta(x^t) &= \frac{1}{\sqrt{\bar{\alpha}^t}}(x^t - \beta^t \sqrt{1 - \bar{\alpha}^t} \epsilon_\theta(x^t, t)), \\ \sigma^t &= \sqrt{\frac{1 - \bar{\alpha}^{t-1}}{1 - \bar{\alpha}^t}} \beta^t. \end{aligned} \quad (3)$$

The neural network will undergo training to optimize the variational upper bound on the negative log-likelihood, which can be estimated via the Monte Carlo algorithm. Consequently, the DDPM would sample from the limit distribution, and then recursively generate samples  $x^t$  using the learned reverse chain. DDPM proposes that it can be trained more effectively by a simplified parameterization schema, which leads to the following objective:

$$\text{Loss}(\theta) = \mathbb{E}_{x^0, \epsilon, t} \left[ \|\epsilon - \epsilon_\theta(x^t, t)\|_2^2 \right], \quad (4)$$

where  $\epsilon_\theta(\cdot)$  is a network estimating noise added to  $x^t$ . Once trained, target variables are first sampled from Gaussian as the input of  $\epsilon_\theta(\cdot)$  to progressively learn the distribution  $p_\theta(x^{t-1}|x^t)$  and denoise  $x^t$  until  $x^0$  is obtained. DDPM decomposes a distribution into a combination of Gaussians, with each step only recovering the simple Gaussian. This capability empowers the model to effectively represent complex distributions, making it suitable for learning the conditional distributions in our tasks.

### C. Neural Process

A neural process is a probabilistic model designed to handle functions and distributions over functions [20]. It generalizes Gaussian processes by leveraging neural networks. Neural processes can learn from a small number of observed data points to make predictions about the entire function. They are particularly useful in scenarios where data is limited or expensive to obtain, such as in few-shot learning tasks. The key idea is to capture the underlying structure of the function space, enabling the model to generate reasonable function values for new input points based on the learned patterns from the available data.

Let's consider a set of input points  $X = \{x_1, x_2, \dots, x_n\}$  and their corresponding target values  $Y = \{y_1, y_2, \dots, y_n\}$ . A neural process consists of two main components: an encoder and a decoder. The encoder takes the input-output pairs  $(X, Y)$  and encodes them into a latent representation  $z$ . Mathematically, the encoding process can be represented as  $Z = \text{Encoder}(X, Y)$ .

The decoder then takes a new input point  $x^*$  and the latent representation  $Z$  to predict the distribution of the corresponding output  $y^*$ . The prediction is formulated as  $p(y^*|x^*, X, Y) = \text{Decoder}(x^*, Z)$ . In many cases, the decoder outputs the mean  $\mu$  and variance  $\sigma^2$  of a Gaussian distribution, so  $p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu(x^*, Z), \sigma^2(x^*, Z))$ . This formulation allows the neural process to not only make point predictions but also quantify the uncertainty associated with those predictions.

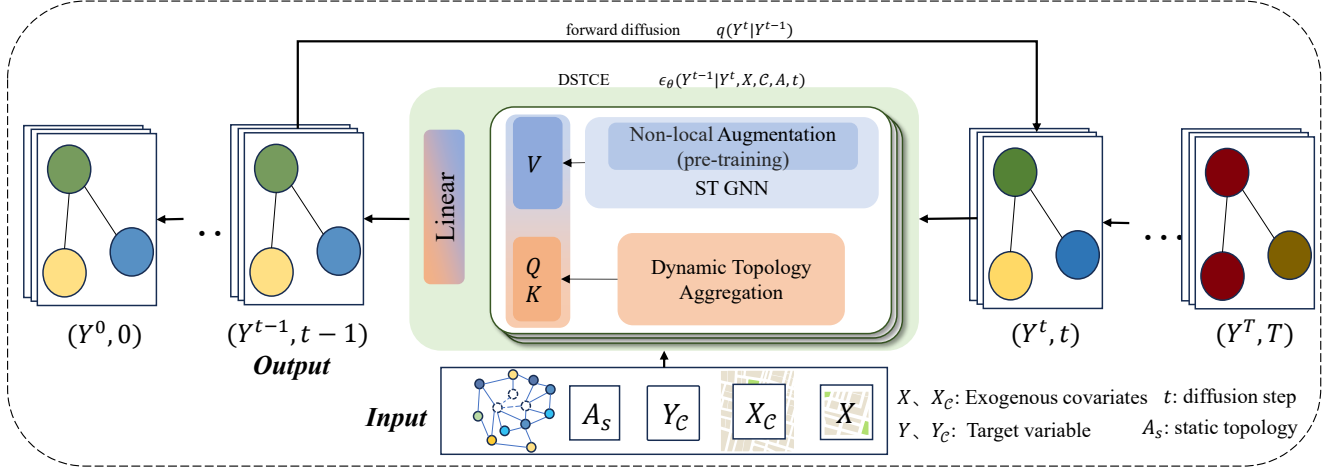


Fig. 2: The proposed DSTCE model framework. Attention learning is organized into multiple layers based on feature scales. Within each layer, STGNN and Non-local Factor Learning module handle the extraction of attention mechanism components, specifically  $V$ , while Dynamic Topology Aggregation incorporates dynamic topology to aggregate context node information into the  $Q, K$  in attention mechanism.

The attention from different layers and feature scales is finally fused through a linear layer, serving as the output for noise prediction.

---

Algorithm 1: Training process of DSTCE

---

Data: The set of context node set  $\mathcal{C}$ , the exogenous covariates  $X$  and the target variables  $Y^0$  of the target node set  $\mathcal{D}$ , the static adjacency matrix  $A_s$ , the number of iterations  $N_{it}$ , the number of diffusion steps  $T$ , noise levels sequence  $\bar{\alpha}_t$

Result: Optimized noise prediction model  $\theta$

- 1 for  $i = 1$  to  $N_{it}$  do
  - 2 Sample  $t \sim \text{Uniform}(\{1, \dots, T\})$ ,  $\epsilon \sim \mathcal{N}(0, I)$ ,
  - 3  $\mathcal{C} \perp \mathcal{D} \sim \text{Dataset}_{\text{train}}$
  - 4  $Y^t \leftarrow \sqrt{\bar{\alpha}^t} Y^0 + \sqrt{1 - \bar{\alpha}^t} \epsilon$
  - 5 Updating the gradient
  - 6  $\nabla_{\theta} \|\epsilon^t - \epsilon_{\theta}(Y^t, X, \mathcal{C}, A_s, t)\|_2^2$
- 

### III. METHODOLOGY

In this section, we propose DSTCE, a diffusion probabilistic model for spatiotemporal extrapolation. As illustrated in 2, the key components are non-local factor learning and dynamic aggregation modules that exploit covariate information and adapt to dynamic scenarios. We introduce the spatiotemporal extrapolation problem, then describe the training and sampling stages, and finally detail the two key modules and attention-based denoising network  $\epsilon$ .

#### A. Spatiotemporal Extrapolation

Spatiotemporal extrapolation is a method of predicting data in unknown spatiotemporal regions based on known spatiotemporal data. We refer to our problem as an extrapolation problem, using known context nodes to infer target variable in target nodes through the learned function  $f(\cdot)$ . Unlike other extrapolations, we also use

exogenous covariates to learn the interaction between target variable and exogenous covariates for extrapolation assistance:

$$\mathcal{C}\{X_{1:L}, Y_{1:L}\}, \mathcal{D}\{X_{1:L}\} \xrightarrow{f(\cdot)} \mathcal{D}\{Y_{1:L}\}. \quad (5)$$

For clarity, we denote the spatiotemporal target data within the target set  $\mathcal{D}\{Y_{1:L}\}$  as  $Y$ , and the spatiotemporal exogenous covariates within target set  $\mathcal{D}\{X_{1:L}\}$  as  $X$ . Furthermore, the target data within the context set  $\mathcal{C}\{Y_{1:L}\}$  is referred to as  $Y_c$ , and the exogenous covariates within the context set  $\mathcal{C}\{X_{1:L}\}$  as  $X_c$ .

#### B. Inductive Diffusion Model Training

In response to the limitation that existing spatiotemporal denoising diffusion probability models [14], [21] cannot be applied to spatiotemporal extrapolation problems, we adopt the conditional diffusion probability model proposed by CSDI [14], utilizing spatiotemporal aggregation information as a condition for guided diffusion model generation. The forward noise addition process is shown as follows:

$$q(Y^t | Y^{t-1}) = \mathcal{N}(Y^t; \sqrt{1 - \beta^t} Y^{t-1}, \beta^t I). \quad (6)$$

The meaning of  $\beta$  in Eq. (6) is consistent with that in Eq. (1), where  $Y$  represents the target spatiotemporal data for the region. While the backward process is modified to incorporate a conditional denoising approach, it is specifically shown as follows:

$$p_{\theta}(Y^{0:T} | Y^T, X, \mathcal{C}, A_s) = \prod_{t=1}^T p_{\theta}(Y^{t-1} | Y^t, X, \mathcal{C}, A_s). \quad (7)$$

The loss function of the conditional diffusion extrapolation model is derived from Eq. (3) and Eq. (4) as follows:

$$L(\theta) = \mathbb{E}_{x^0, \epsilon, t} \left[ \|\epsilon - \epsilon_{\theta}(Y^t, X, \mathcal{C}, A_s, t)\|_2^2 \right]. \quad (8)$$

---

**Algorithm 2: Extrapolation process with DSTE**


---

Data: The set of context node set  $\mathcal{C}$ , and the exogenous covariates  $X$  of the target node set  $\mathcal{D}$ , the static adjacency matrix  $A_s$ , the number of diffusion steps  $T$ , the optimized noise prediction model  $\epsilon_\theta$

Result: Unobserved extrapolation target values  $Y^0$

- 1 Sample  $Y^T \sim \mathcal{N}(0, I)$ ,  $\mathcal{C} \sim \text{Dataset}_{\text{test}}$
  - 2 for  $t = T$  to 1 do
  - 3  $\mu_\theta(Y^t, X, \mathcal{C}, A_s, t) \leftarrow$   
 $\frac{1}{\sqrt{\alpha^t}}(Y^t - \beta^t \sqrt{1 - \alpha^t} \epsilon_\theta(Y^t, X, \mathcal{C}, A_s, t))$
  - 4 Reverse denoising  
 $Y^{t-1} \leftarrow \mathcal{N}(\mu_\theta(Y^t, X, \mathcal{C}, A_s, t), \sigma_t^2 I)$
- 

The improved diffusion extrapolation model integrates the target node exogenous covariates  $X$ , the context node set  $\mathcal{C}$ , and an additional condition represented by the static adjacency matrix  $A_s$ , which captures the static topological relationships. The entire reverse process involves predicting the added noise for the extrapolation target, with the goal of restoring the original information of the noisy sample. As a result,  $\theta$  is often referred to as the noise prediction model.

The training and extrapolation processes of the model are illustrated in the framework diagram shown in Fig. 2. Specifically, during the training process, we start by selecting a context node set  $\mathcal{C}$  and a target node set  $\mathcal{D}$  from a pre-divided training dataset, ensuring that the two sets do not overlap. Next, we introduce varying degrees of Gaussian noise [18] to the target variables  $Y^0$  in the designated target node set  $\mathcal{D}$ , resulting in  $Y^t$ . Subsequently, we use the perturbed  $Y^t$ , diffusion time step  $t$ , exogenous covariates  $X$ , the context node set  $\mathcal{C}$ , and the static topology between nodes  $A_s$  as inputs. Through the denoising module, we aim to learn the conditional distribution of the target variables by removing the noise added to  $Y^t$ . The extrapolation process is the reverse of the denoising process. At this stage, the target node set  $\mathcal{D}$  is already specified. We initialize the target variables at the designated positions with pure Gaussian noise  $Y^T$ . The remaining inputs follow a similar structure to the training process. Through iterative denoising steps, we progressively restore  $Y^T$  to the extrapolated target values  $Y^0$ .

We are not making substantial alterations to the overall training process of the denoising diffusion model. This decision is primarily influenced by our emphasis on designing a denoising module tailored for extrapolation problems. Our research aims to facilitate inductive reasoning by enabling the denoising model  $\epsilon_\theta$  to learn the interdependence between conditions and target recovery.

### C. Denoising Neural Network

The Denoising Neural Network  $\epsilon_\theta$  is a key part in our model, which makes DSTE have generative ability to transform  $Y^T$  to  $Y^0$  by inferring added noise  $\epsilon_t$  at

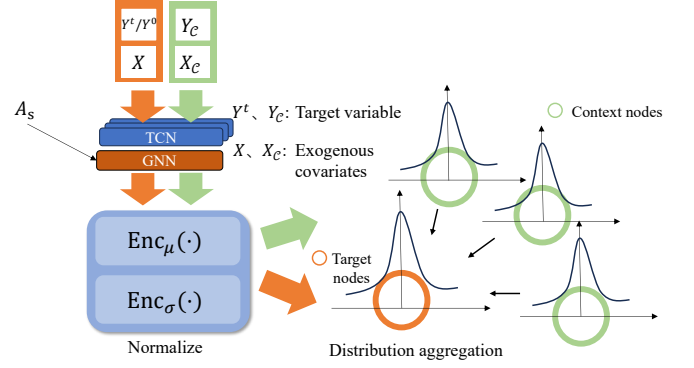


Fig. 3: Non-local Factor Learning. Converting the output of the TCN and GNN into a normal distribution through  $\text{Enc}_\mu(\cdot)$  and  $\text{Enc}_\sigma(\cdot)$ . These transformation modules enable the representation of spatiotemporal features as probabilistic distributions, capturing both the expected values and their associated uncertainties.

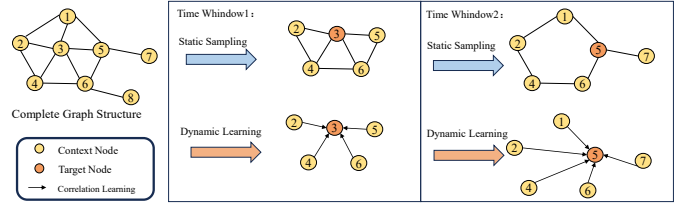


Fig. 4: Hybrid modeling overview. Red (target) and yellow (context) nodes form the spatiotemporal graph. Arrows show dynamic learning; lines show static topology. Static and dynamic modules together enable comprehensive spatial feature representation.

every diffusion step  $t$ . It mainly consists of three modules. The Non-local factor learning aims to capture non-regional factors within spatiotemporal data to enhance model generalization. The Dynamic Topology Aggregation Module is designed to capture complex and time-varying correlations between nodes in real-world scenarios. The Gradient Guided Attention Module extracts and integrates relationships among features from the former two modules for the extrapolation denoising diffusion model.

1) Non-local Factor Learning Module (Pre-training) To minimize information loss from context nodes during aggregation, we propose a non-local factor learning module. This module maps the features of target and context nodes into a Gaussian-form non-local factor via a neural network. The design aims to comprehensively capture global dependencies among nodes and incorporates uncertainty in node interactions. Through pre-training, this module learns cross-spatiotemporal non-local interaction patterns.: The raw data consists of target variables and exogenous covariates for all nodes, denoted as  $\{Y, X\} \in [\mathcal{C}, \mathcal{D}]$ . The input is first processed by a Spatio-Temporal Graph Neural Network (STGNN), which comprises a stacked Temporal Convolutional Network (TCN) and a Graph Neural Network (GNN). The TCN employs convolutional layers instead of recurrent ones to efficiently capture long-term dependencies in sequential data [22], while the



GNN extracts spatial features based on the static topology  $A_S$ . This process provides the foundational spatiotemporal representations required for the pre-training of the non-local factor learning module.

$$\begin{aligned}\mu_F &= \text{Enc}_\mu(\text{GNN}(\text{TCN}([Y^t \| X]), A_S)), \\ \sigma_F &= \text{Enc}_\sigma(\text{GNN}(\text{TCN}([Y^t \| X]), A_S)).\end{aligned}\quad (9)$$

After processing through GNN and TCN layers, the target node set (D) and context nodes (C) are mapped through linear layer normalization modules  $\text{Enc}_\mu(\cdot)$  and  $\text{Enc}_\sigma(\cdot)$  to form their respective node mappings:

$$\begin{aligned}\mu_{F_C} &= \text{Enc}_\mu(\text{GNN}(\text{TCN}([Y_C \| X_C]), A_S)), \\ \sigma_{F_C} &= \text{Enc}_\sigma(\text{GNN}(\text{TCN}([Y_C \| X_C]), A_S)),\end{aligned}\quad (10)$$

here,  $\text{Enc}_\mu(\cdot)$  and  $\text{Enc}_\sigma(\cdot)$  output the mean  $\mu$  and variance  $\sigma$ , forming the Gaussian non-local factor  $F_{[C,D]} \sim \mathcal{N}(\mu, \sigma^2)$ . Leveraging the pre-trained factors  $F_{[C,D]}$  and the static adjacency matrix  $A_S$ , we perform non-local factor fusion between target nodes (D) and context nodes (C) as follows:

$$\begin{aligned}\bar{\mu}_F &= \bar{\sigma}_F^2 \left( \mu_F / \sigma_F^2 + \sum_{n \in \mathcal{N}_1^C(m)} A_S \mu_{F_C} / \sigma_{F_C}^2 \right), \\ \bar{\sigma}_F^2 &= \left[ \sigma_F^{-2} + \sum_{n \in \mathcal{N}_1^C(m)} (\sigma_{F_C} / A_S)^{-2} \right]^{-1},\end{aligned}\quad (11)$$

where  $\mu_F$  and  $\sigma_F$  are the pre-trained factor parameters derived from the encoding process, and  $\bar{\mu}_F$ ,  $\bar{\sigma}_F$  represent the comprehensively fused non-local factor distribution at the target node.  $\mathcal{N}_1^C(m)$  denotes the first-order neighborhood of context nodes surrounding the target node  $m$ .

$$\begin{aligned}\log p(Y|C, X, A) &\geq \underbrace{\mathbb{E}_q[\log p_{\text{np}}(Y|C, F, X, A)]}_{\text{(b) log-likelihood Term}} \\ &\quad - \underbrace{\text{KL}(q(F|C \cup D) \| p(F|C))}_{\text{(c) KL Regularization}}\end{aligned}\quad (12)$$

During the pre-training process, we utilize the aggregated information from target node set ( $D, Y = Y^0$ ) and context nodes (C) as the prior distribution, while using pure noise as target nodes  $D, Y \sim \mathcal{N}(0, 1)$  and the aggregated information from context nodes (C) as the posterior distribution. The KL divergence between these distributions serves as a constraint, while the target values of the posterior distribution are trained against  $Y^0$  using negative log-likelihood. This approach ultimately yields our pre-trained model. The loss function of the pretraining phase is illustrated in Eq. (12).

In the pre-training phase, we establish a framework where the aggregated information from the target node set ( $D, Y = Y^0$ ) and context nodes (C) serves as the prior distribution. Concurrently, we employ pure Gaussian noise for target nodes  $D, Y \sim \mathcal{N}(0, 1)$  combined with

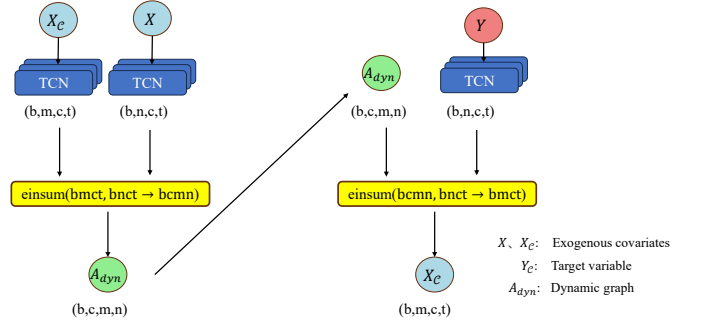


Fig. 5: Target and context node features are projected to compute a dynamic adjacency matrix via tensor multiplication. This matrix aggregates context predictions to targets, followed by normalization and projection to update target predictions, enabling adaptive, time-varying relationship learning.

aggregated context node (C) information to form the posterior distribution. We constrain the model using the KL divergence between these distributions, while simultaneously training with negative log-likelihood between the posterior distribution's target values and  $Y^0$ . This comprehensive approach results in our robust pre-trained model. This process establishes an implicit association between exogenous covariates and target variables (see [2] for proof of the formulas).

2) Dynamic Topology Aggregation Module: To ensure the model's inductive learning capability, we employ partial graph sampling across different time windows from the complete spatiotemporal graph during training, while simultaneously learning the dynamic relationships between target nodes and context nodes within the sampled subset, as illustrated in Figure 4

However, we have noted that in real-world scenarios, the correlation between regions is not solely determined by their geographical distances. Various factors such as weather conditions, terrain, construction of transportation infrastructure, and traffic policies can contribute to complex interconnections between regions. This correlation may not be linearly related to geographic distance, and in some cases it may exhibit exponential decay with an increase in distance, a phenomenon commonly referred to as the distance decay effect. Additionally, it is sometimes observed that points farther apart exhibit stronger correlations compared to other nodes in the network.

Accounting for this relationship is crucial in spatiotemporal extrapolation tasks. While statistical methods informed by prior knowledge may seem reliable, applying such methods in dynamically complex scenarios is often impractical. On one hand, different application scenarios may require distinct statistical approaches, necessitating extensive research to devise a theoretically sound statistical method for newly emerging contexts, which might lack timeliness in urgent situations. On the other hand, the correlation structure in spatiotemporal contexts may change over time, and failure to account for such changes in statistical methods could result in the failure of correlation predictions.

To address these challenges, we propose a dynamic topology aggregation module that combines static sampling with dynamic graph learning for extrapolation tasks. This innovative approach leverages the efficiency of static graph sampling while incorporating the flexibility of dynamic graph learning. The module's core functionality is to learn time-varying dynamic correlations between target and context nodes, and efficiently aggregate contextual information onto the target nodes.

$$A_{D_{yn}} = (X \star \mathcal{K}_1)(X_C \star \mathcal{K}_1). \quad (13)$$

Eq. (13) illustrates the aggregation process of our dynamic graph, accomplished by employing the same channel convolution  $\star \mathcal{K}_1$  as the target node and the exogenous covariates feature extraction module of the context. This convolution operation learns the co-variant feature extraction between the target node and the context nodes, resulting in the computation of the dynamic correlation matrix  $A_{D_{yn}} \in \mathbb{R}^{M \times N \times dx'}$ . Here,  $M$  and  $N$  represent the quantities of target and context nodes in the inference process, and  $dx'$  denotes the dimension of features after  $\star \mathcal{K}_1$ .

$$H_{DT} = (Y_C \star \mathcal{K}_2)A_{D_{yn}}. \quad (14)$$

Eq. (14) further illustrates the process of aggregating the dynamic graph matrix of the context nodes onto the target node after feature extraction.  $H_{DA} \in \mathbb{R}^{M \times dy' \times L}$  represents the aggregated values of the dynamic context, where  $dy'$  denotes the values after channel convolution, and  $L$  represents the time length.

3) Gradient Guided Attention Module: The denoising attention guidance module plays a crucial role in the extrapolation denoising diffusion model. The main purpose of this part is to extract the relationship between the target node features and the context-aggregated features obtained from the Non-local Factor Learning module as well as the Dynamic Topology Aggregation module. For the sake of clarity, we use the predicted target  $Y^t$  at the previous diffusion step  $t$  as the input of the attention module [23]. The input is processed by the above two modules respectively to form  $H_{DT}$  and  $H_F$ . Samples of  $H_F$  are then drawn from the normal distribution  $\mathcal{N}(\bar{\mu}_F, \bar{\sigma}_F^2)$ . Then, as shown in 6(a), a cross-attention block is utilized to measure the dependencies between  $H_{DT}$  and  $H_F$ . The following equations illustrate the process:

$$\begin{aligned} Q_{ca} &= H_F \cdot W_{ca}^Q, \\ K_{ca} &= H_F \cdot W_{ca}^K, \\ V_{ca} &= H_{DT} \cdot W_{ca}^V, \end{aligned} \quad (15)$$

$$H_{ca} = \text{softmax} \left( \frac{Q_{ca} K_{ca}^T}{\sqrt{d_h}} V_{ca} \right). \quad (16)$$

Here,  $H_{ca}$  represents the results, and  $W_{ca}^Q, W_{ca}^K, W_{ca}^V$  are matrices that can be learned. To enhance the spatial relationship mining in Non-local Factor information, we

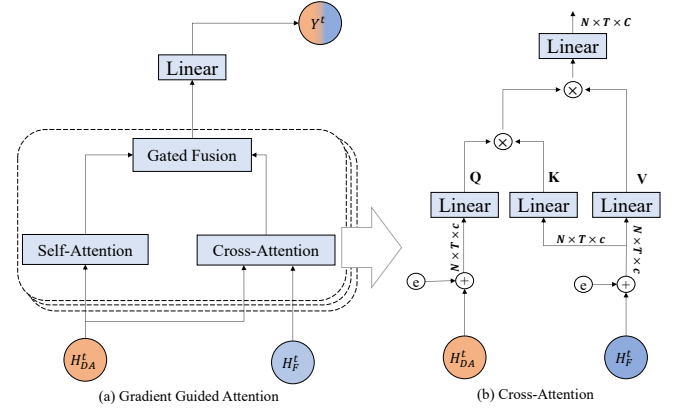


Fig. 6: Gradient Guided Attention Module, The diagram shows the overall aggregation module (a) and cross-attention module (b). The overall aggregation module integrates self-attention and cross-attention through a gated fusion mechanism to process the input data with shape  $N \times T \times c$  where  $c$  is feature channel and generate the output  $Y^{t-1}$  for the next diffusion step. The cross-attention module measures the dependencies between  $H_F$

and  $H_{DT}$  to obtain  $H_{ca}$

employ self-attention to obtain as follows (similar to equation 16):

$$\begin{aligned} Q_{sa} &= H_F \cdot W_{sa}^Q, \\ K_{sa} &= H_F \cdot W_{sa}^K, \\ V_{sa} &= H_F \cdot W_{sa}^V. \end{aligned} \quad (17)$$

Finally, a gated fusion mechanism is proposed to integrate the outputs  $Y^{t-1}$ :

$$Y^{t-1} = \text{Linear}(\gamma \cdot H_{sa} + (1 - \gamma) \cdot H_{ca}), \quad (18)$$

where  $Y^{t-1}$  is the Gradient Guided Attention output and serves as the input in the next diffusion step.  $\gamma$  is a learnable parameter used to balance the self-attention outputs and cross-attention outputs.

#### D. Joint Variational Lower Bound for Factor-Augmented Diffusion Models

To integrate extrapolated factor learning into the diffusion model in a principled manner, we employ a variational Bayesian framework. Specifically, we reformulate the training objective by incorporating the learned non-local factors into the evidence lower bound (ELBO) of the diffusion process. The resulting variational lower bound is given by:

$$\begin{aligned} \log p(Y|C, X, A) &\geq \underbrace{\mathbb{E}_q[\log p_{\text{diff}}(Y|C, F, X, A)]}_{\text{(a) Diffusion Model Term}} \\ &+ \underbrace{\mathbb{E}_q[\log p_{\text{np}}(Y|C, F, X, A)]}_{\text{(b) Neural Process Term}} \\ &- \underbrace{\text{KL}(q(F|C \cup D) \| p(F|C))}_{\text{(c) KL Regularization}} \end{aligned} \quad (19)$$

The formula presents the variational lower bound (Evidence Lower Bound, ELBO) for the joint log marginal

likelihood of observations  $Y$ , under the scenario where a diffusion model and a neural process share a latent variable space  $R$ . The interpretation of each term is as follows:

1) Diffusion Model Term: This term represents the expected log-likelihood of the observed data  $Y$  given the context  $C$ , latent variable  $R$ , input  $X$ , and optional condition  $A$ , as modeled by the diffusion model. It measures how well the diffusion model explains the observations under the current latent representation.

2) Neural Process Term: This term quantifies the modeling capacity of the neural process for the same observations, conditioned on the shared latent variable  $R$ . By sharing  $R$ , the neural process can leverage richer context information to enhance generalization.

3) KL Regularization Term: This term is the Kullback-Leibler (KL) divergence between the variational posterior  $q(R|C \cup D)$  and the prior  $p(R|C)$ . It acts as a regularizer, constraining the latent variable distribution to stay close to the prior, thus preventing overfitting and promoting better generalization.

The objective of optimizing this ELBO is to jointly improve the explanatory power of both the diffusion model and the neural process with respect to the observed data, while maintaining a reasonable latent variable distribution. By sharing the latent space  $R$ , both models can complement each other, fully utilizing both context and observation information to enhance generative and generalization capabilities.

Proof 3.1: Proof: To achieve joint training of diffusion models and neural processes, this chapter introduces a shared noise latent variable space and defines the joint interaction factor  $F = (F_{\text{diff}}, F_{\text{np}})$ . Here,  $F_{\text{diff}}$  represents the interaction factor of the diffusion model, and  $F_{\text{np}}$  represents the interaction factor of the neural process. During training,  $F$  serves simultaneously as the output of the neural process and as the conditional input of the diffusion model, thereby achieving joint optimization of both models.

According to the joint probability decomposition, we have:

$$p(Y|C, X, A) = \int p_{\text{diff}}(Y, F_{\text{diff}}|C, F, X, A) p_{\text{np}}(F_{\text{np}}|C, X, A) dF \quad (20)$$

Introducing the variational distribution  $q(F|C \cup D)$  to approximate the posterior distribution, and according to Jensen's inequality:

$$\log \mathbb{E}_q[f(x)] \geq \mathbb{E}_q[\log(f(x))] \quad (21)$$

We obtain:

$$\begin{aligned} \log p(Y|C, X, A) &= \log \mathbb{E}_q \left[ \frac{p_{\text{diff}}(Y, F_{\text{diff}}|C, F, X, A) p_{\text{np}}(F_{\text{np}}|C, X, A)}{q(F|C \cup D)} \right] \\ &\geq \mathbb{E}_q \left[ \log \frac{p_{\text{diff}}(Y, F_{\text{diff}}|C, F, X, A) p_{\text{np}}(F_{\text{np}}|C, X, A)}{q(F|C \cup D)} \right] \end{aligned} \quad (22)$$

During joint training, we assume  $F_{\text{diff}} = F_{\text{np}} = F$ , i.e., the interaction factor is shared between the diffusion model and neural process. In this case, equation (4.24) can be further simplified to:

$$\begin{aligned} \log p(Y|C, X, A) &\geq \mathbb{E}_q[\log p_{\text{diff}}(Y|C, F, X, A)] \\ &\quad + \mathbb{E}_q[\log p_{\text{np}}(Y|C, F, X, A)] \\ &\quad - \text{KL}(q(F|C \cup D) \| p(F|C)) \end{aligned} \quad (23)$$

#### IV. EXPERIMENTS

In this section, we first introduce the datasets, baselines, evaluation metrics, and experiment settings. Subsequently, to validate the effectiveness of our model, we present the following questions and address them in the following sections.

- Q1: Can our proposed extrapolation model outperform other baseline methods and achieve state-of-the-art results in various environments?
- Q2: What is the effectiveness of the components in our model, such as DS, DT, and GAttn?
- Q3: Is the model sensitive to hyper-parameters and prone to over fitting?
- Q4: Is our model robust in extrapolated regions, and can it maintain good performance in different unseen areas, including cross-city scenarios?

##### A. Dataset

Beijing [24] contains air quality indexes (AQI) from 35 stations and district-level meteorological attributes. We aim to extrapolate the AQI of PM2.5, PM10, and NO2, using meteorological attributes such as temperature, humidity, pressure, wind speed, wind direction, and weather as exogenous covariates.

London is similar to the Beijing dataset, with the same data format and 24 stations. During training, we input hourly data with a time period of 24 hours. It is worth mentioning that the final dataset used in London and Beijing is a combination of AQI dataset and exogenous covariate dataset represented by grid. The exogenous covariate of each AQI monitoring station is the corresponding content within the current grid.

We use datasets in grid form as covariates mainly due to the following reasons: 1. Monitoring covariates such as temperature and humidity is relatively easy, and detection often only requires simple physical principles to detect; AQI data, on the other hand, requires complex optical sensing or chemical detection techniques. Detectors are expensive and require frequent maintenance. In addition, due to various reasons such as price, the data of these covariates is relatively rich, making it easier and more practical to infer discrete missing information from the known rich information.

IntelLab [25], this dataset contains information about data collected from 54 sensors deployed in the Intel Berkeley Research lab between February 28th and April 5th, 2004. Temperature is in degrees Celsius. Humidity is temperature corrected relative humidity, ranging from 0-100%. Light is in Lux. Voltage is expressed in volts, ranging from 2-3, and is highly correlated with temperature [25]. We multiply the obtained voltage value by



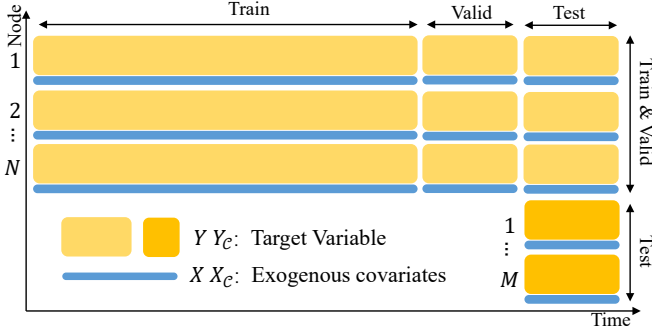


Fig. 7: During training/validation, each iteration selects mutually exclusive target and context sets from the  $N$  training nodes. In testing, context nodes come from these  $N$  nodes, while target nodes are  $M$  distinct test nodes, all within the test interval.

TABLE II: Hourly Statistics for Beijing AQI Data Sets.

| Data Type            | Air-Quality   |                                  |                                 |
|----------------------|---|----------------------------------|---------------------------------|
| Target variables     | PM2.5( $\mu\text{g}/\text{m}^3$ )   | PM10( $\mu\text{g}/\text{m}^3$ ) | NO2( $\mu\text{g}/\text{m}^3$ ) |
| Exogenous covariates | Weather, Temperature( $^{\circ}\text{C}$ ), Pressure(hpa), Humidity(%), Wind speed(m/s), Wind direction |                                  |                                 |
| Subareas             | 35  |                                  |                                 |
| Duration             | 2017/5/1 to 2018/4/30   |                                  |                                 |
| Mean $\pm$ Std       | 84.65 $\pm$ 81.20   | 112.85 $\pm$ 101.17              | 52.11 $\pm$ 37.02               |

100 to represent Voltage100. Due to the misalignment of the original data’s time, we have reduced the average processing interval of the data to 10 minutes, and the input time period is 12 hours. Fill in the missing value with 0 and provide a prompt when calculating the loss during training, indicating that the information does not enter the input. Here, we conducted experiments on all four variables as target values, and only presented the experimental results of voltage as the extrapolation target to demonstrate the potential of our system for more practical physical quantities.

## B. Baselines

In establishing our baseline models, we select classical statistical and machine learning methods alongside representative neural network approaches. We opt for Inverse Distance Weighting (IDW) [26], a classical interpolation technique widely used for spatial extrapolation tasks. Additionally, classical machine learning baselines include k-nearest neighbors (KNN) [27] and random forest (RF) [28]. Within our machine learning repertoire, we incorporate XGBoost [29], an algorithm that leverages a gradient-boosting framework to effectively amalgamate predictions from multiple weak learners.

Among neural network methods, we employ both deterministic and probabilistic models. For deterministic approaches, we include ADAIN [30], which combines MLP and RNN architectures to aggregate data features, and Vision Transformer (ViT) [31], which applies self-attention

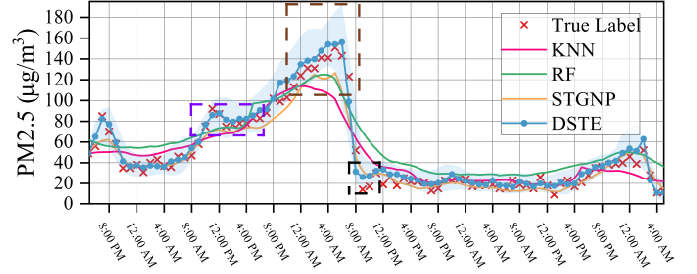


Fig. 8: PM2.5 extrapolation results on the Beijing dataset from April 3 to 7, 2017. The blue regions and lines represent our proposed method, while the red crosses indicate the ground truth. The dashed box highlights that our method demonstrates superior spatiotemporal extrapolation capability compared to other methods.

mechanisms to capture long-range spatiotemporal dependencies. For probabilistic methods, we utilize STGNP [2], which provides results as probability distributions rather than deterministic values. We also incorporate generative approaches including the GAN-based method GPNv2 [32] and diffusion-based methods PriSTI [3] and CaPaint [33].

## C. Evaluation Strategy and Hyperparameters

Our experimental setup closely follows the design of [2]. As illustrated in Fig. 7, prior to training, three nodes are randomly selected from the dataset to form the final test set, and they are excluded from the training set. The remaining nodes are categorized into target nodes and conditional nodes during each training iteration, with  $N$  nodes chosen as targets and  $M$  nodes as conditions, ensuring that  $3M = 7N$ . In the final testing phase, all  $(N + M)$  nodes are used as conditions to evaluate the three nodes selected before training, which are not part of the training set. The dataset is temporally divided into three segments: an 80% training set, a 10% validation set, and a 10% test set.

In the experimental process, we set the diffusion training parameters, including the diffusion step size  $T$  as 100,  $\beta^0$  and  $\beta^T$  as 0.0001 and 0.2, respectively. We adopted the quadratic schedule for other noise levels following [14], which is formalized as:

$$\beta_t = \left( \frac{T-t}{T-1} \sqrt{\beta_1} + \frac{t-1}{T-1} \sqrt{\beta_T} \right)^2. \quad (24)$$

The embedding of diffusion time and temporal encoding is realized through sine and cosine embeddings, building upon prior studies [14], [15]. The learning rate decreases from 0.001 to 0.0001 at 75% of the total epochs. The internal feature channels of each layer within the attention module are 16, 32, 64, 128. After attention computation, the channels are merged, and the merged outputs are passed through two linear layers to produce the final outputs.

## D. Results

1) Overall Performance (Q1): We evaluate our proposed method on the Beijing dataset, which includes three

TABLE III: Performances of DSTE and the baselines.

| Model   | PM2.5        |              |             | PM10         |              |             | NO2          |              |             | PM2.5 (London) |             |             | Voltage100 (IntelLab) |              |             |
|---------|--------------|--------------|-------------|--------------|--------------|-------------|--------------|--------------|-------------|----------------|-------------|-------------|-----------------------|--------------|-------------|
|         | MAE          | RMSE         | MAPE        | MAE          | RMSE         | MAPE        | MAE          | RMSE         | MAPE        | MAE            | RMSE        | MAPE        | MAE                   | RMSE         | MAPE        |
| IDW     | 41.12        | 52.13        | 1.23        | 78.71        | 130.75       | 0.88        | 27.11        | 37.51        | 0.99        | 8.39           | 12.43       | 1.45        | 18.89                 | 23.46        | 0.09        |
| KNN     | 28.7         | 40.78        | 0.67        | 70.21        | 108.2        | 0.7         | 25.45        | 33.21        | 1.08        | 16.96          | 22.52       | 0.64        | 23.28                 | 30.01        | 0.11        |
| RF      | 20.05        | 33.35        | 0.47        | 47.43        | 91.08        | 0.53        | 21.04        | 26.42        | 1.00        | 16.94          | 22.47       | 0.64        | 21.54                 | 29.73        | 0.10        |
| XGB     | 16.30        | 25.62        | 0.37        | 41.08        | 78.66        | 0.41        | 15.79        | 20.97        | 0.61        | 14.57          | 20.07       | 0.49        | 19.66                 | 37.19        | 0.10        |
| Vit     | 18.77        | 34.37        | 0.39        | 34.86        | 54.61        | 0.31        | 13.96        | 18.22        | 0.38        | 5.79           | 6.21        | 0.57        | 12.13                 | 15.74        | 0.08        |
| ADAIN   | 17.87        | 29.24        | 0.36        | 34.2         | 61.29        | 0.36        | 15.73        | 21.3         | 0.58        | 8.41           | 10.26       | 0.55        | 12.23                 | 16.55        | 0.08        |
| STGNP   | <u>15.74</u> | 30.26        | 0.31        | <u>32.26</u> | 52.04        | <u>0.28</u> | 13.38        | 19.72        | 0.41        | <u>3.56</u>    | <u>4.72</u> | 0.63        | 15.31                 | 17.12        | 0.06        |
| GPNv2   | 23.30        | 32.01        | 0.42        | 39.49        | 50.80        | 0.42        | 19.88        | 24.76        | 0.56        | 8.29           | 10.29       | 0.66        | 19.13                 | 23.47        | 0.09        |
| PriSTI  | 17.54        | 30.12        | 0.37        | 35.76        | 54.29        | 0.38        | 14.73        | 18.83        | 0.42        | 6.01           | 8.78        | 0.67        | 11.09                 | 14.19        | 0.11        |
| CaPaint | 16.61        | <u>28.10</u> | <u>0.29</u> | 32.66        | <u>50.09</u> | 0.30        | <u>13.23</u> | <u>18.06</u> | <u>0.35</u> | 4.42           | 5.29        | <u>0.54</u> | <u>10.63</u>          | <u>13.87</u> | <u>0.05</u> |
| DSTE    | 14.14        | 23.24        | 0.24        | 22.78        | 46.40        | 0.23        | 10.28        | 14.64        | 0.30        | 3.11           | 4.07        | 0.60        | 7.94                  | 10.28        | 0.04        |

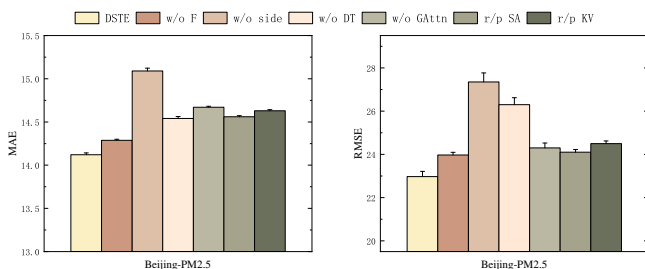


Fig. 9: Ablation Study. DSTE is the full model. ‘w/o F’ removes Non-local Factor Learning;

‘w/o side’ drops covariate fusion; ‘w/o DT’ replaces dynamic topology with distance matrices; ‘w/o GAttn’ removes gradient-guided attention; ‘r/p SA’ uses self-attention instead; ‘r/p KV’ sets Non-local output as  $Q$  and dynamic topology output as  $KV$ .

target variables: PM2.5, PM10, and NO2, along with exogenous covariates such as wind direction, wind speed, and rainfall. In the comparative methods, all approaches utilize exogenous covariates as auxiliary information. In the experimental results, we observe that neural network methods outperform traditional statistical and machine learning methods in terms of mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

Partial results, as shown in Fig. 8, demonstrate the superior inference accuracy of our model around the peak in the violet and brown boxes compared to other methods of machine learning and neural networks. Among the early machine learning methods, KNN, IDW, and RF have their own characteristics. KNN is known for its simplicity in handling data based on proximity in the feature space. IDW is often used in spatial interpolation scenarios with its unique distance - weighted approach. RF utilizes multiple decision trees for better generalization.

XGBoost, a top-performing method in machine learning, tends to exhibit relatively stable changes in the temporal dimension, with a limited representation of time features. For conciseness, the fitting results of some methods during this time interval are not illustrated.

Among neural network methods, we use both deterministic and probabilistic models. In deterministic mod-

els, ADAIN is considered a deterministic method which combines RNNs and fully connected layers.

ViT uses the Vision Transformer architecture, which applies the self-attention mechanism from natural language processing to image-like data by treating spatiotemporal patches as sequences of tokens, enabling it to capture long-range dependencies and global contextual information effectively.

In probabilistic models, STGNP is currently considered the state-of-the-art method. However, our proposed method, DSTE, is outperforming previous methods across all metrics and significantly outshines STGNP on some data subsets.

GPNv2 is a GAN-based method that uses dual attention mechanisms (Spatiotemporal Image Correlation attention and Channel-Spatial attention) to mitigate echo attenuation in radar precipitation nowcasting.

PriSTI introduces a conditional diffusion framework for spatiotemporal imputation that extracts spatiotemporal dependencies as global priors and employs geographic-aware noise estimation to transform random noise into missing values

CaPaint introduces a causal spatiotemporal framework that identifies causal regions via Vision Transformer attention and performs diffusion-based inpainting on non-causal areas to enhance model performance and interpretability.

Our proposed method, DSTE, outperforms previous methods in all metrics. This improvement is attributed to the powerful distribution-capturing capabilities of the diffusion probability model used in our method. The results are provided in TABLE III. Underlined values indicate the best performance among the other models.

Simultaneously, we employ the Continuous Ranked Probability Score (CRPS) as our evaluation metric. CRPS assesses the compatibility of the estimated probability distribution with the observed value. The calculation details of CRPS are introduced as follows: For a missing value  $x$  with an estimated probability distribution  $D$ , CRPS measures the compatibility of  $D$  and  $x$ , defined as the integral of the quantile loss  $\Lambda_\alpha$ :

$$CRPS(D^{-1}, x) = \int_0^1 2\Lambda_\alpha(D^{-1}(\alpha), x) d\alpha, \quad (25)$$

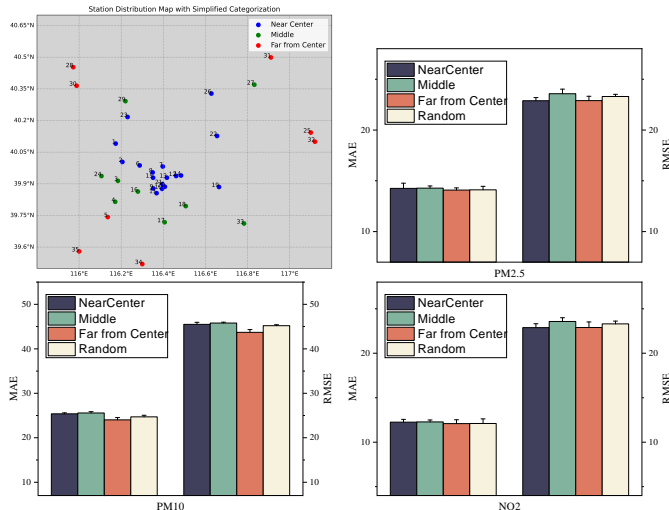


Fig. 10: Regions Evaluation. We categorizes Beijing air quality monitoring stations into three regions—Near Center (blue), Middle (green), and Far from Center (red)—using a normal distribution approach. It calculates mean distance and standard deviation, computes Z-scores, and assigns stations to regions based on predefined thresholds. The resulting map visually displays the spatial distribution, with each region represented by a distinct color and concentric circles indicating standard deviations.

TABLE IV: Evaluation Metrics for Different Regions

| Region          | MAE   | PM2.5 |      |      |
|-----------------|-------|-------|------|------|
|                 |       | RMSE  | MAPE | CRPS |
| Baseline        | 15.74 | 30.26 | 0.31 | 0.43 |
| Near Center     | 14.06 | 22.88 | 0.23 | 0.41 |
| Middle          | 14.23 | 23.57 | 0.25 | 0.41 |
| Far from Center | 14.09 | 22.90 | 0.24 | 0.40 |
| Random          | 14.11 | 23.39 | 0.24 | 0.41 |

where  $\alpha \in [0, 1]$  represents the quantile levels,  $D^{-1}(\alpha)$  is the  $\alpha$ -quantile of distribution  $D$ , and  $I$  is the indicator function. The quantile loss function  $\Lambda_\alpha$  is defined as:

$$\Lambda_\alpha(D^{-1}(\alpha), x) = (\alpha - I_{x < D^{-1}(\alpha)})(x - D^{-1}(\alpha)), \quad (26)$$

Following [2], as our distribution of missing values is approximated by generating 100 samples, we compute quantile losses for discretized quantile levels with 0.05 ticks:

$$CRPS(D^{-1}, x) \approx \frac{1}{19} \sum_{i=1}^{19} 2\Lambda_{i \times 0.05}(D^{-1}(i \times 0.05), x). \quad (27)$$

We compute CRPS for each estimated missing value and use the average as the evaluation metric, formalized as:

$$CRPS(D, \tilde{X}) = \frac{\sum_{\tilde{x} \in \tilde{X}} CRPS(D^{-1}, \tilde{x})}{|\tilde{X}|}. \quad (28)$$

2) Ablation Study (Q2): To assess the contribution of each component to the overall performance of our model and address Q2, we conducted an ablation study, and the results are presented in Fig. 9. In each study, we modified the corresponding part while keeping other settings unchanged.

Effect of Non-local Factor: We remove the Non-local Factor Learning module, retaining only STGNN, and use its output directly as input for attention mechanisms  $Q$  and  $K$ , with all other modules unchanged (w/o F). The results of the ablation study indicate a significant performance decline when the module is absent. This suggests that utilizing distribution for information aggregation is effective in extrapolation tasks. Effect of side encoding: As the Non-local factor learning module includes  $Enc_{side}$ , i.e., the covariates encoding module, we systematically test the performance of this module by removing  $Enc_{side}$  while keeping other modules unchanged (w/o side). The results indicate a performance decrease when  $Enc_{side}$  is removed, but the magnitude of the decline does not exceed that observed in the without that case. This simultaneously demonstrates the effectiveness of  $Enc_{side}$  in conjunction with the remaining parts of Non-local factor learning module.

Effect of Dynamic Topology Aggregation: To validate the effectiveness of our dynamic information topology, we conduct experiments by removing the dynamic graph learning module and only using the static topology for information aggregation in the DT module (w/o DT). The experimental results show that without the extraction of dynamic features, the performance of the module decreases to some extent.

Effect of GAttn: In this section, we investigate the effects of different attention mechanisms. First, we conduct ablation experiments by removing the entire attention module to verify the effectiveness of our proposed gradient-guided attention mechanism (w/o GAttn). Experimental results demonstrate that the attention module successfully learns the relevance between the aggregated context nodes information and the target nodes, and utilizes the learned information to better eliminate noise. Next, we experiment with the allocation of  $QKV$  in the attention module. We compare the performance of self-attention applied only to distributed aggregated information (r/p  $SA$ ) and the performance of using the distributed aggregated information as  $Q$  and dynamical topological aggregated information as  $KV$  (r/p  $KV$ ). Experimental results show that the original GAttn setting contributes the most to the performance in these settings, indicating that the query subject of the attention mechanism should be the distributed aggregated information of the target node. While using aggregated topological aggregated information as the query content also provides information, it should not be overly incorporated.

3) Hyperparameter Study (Q3): We conduct experiments to assess the performance of DSTE under various hyper-parameter settings. Initially, we focus on the experimentation with the number of feature channels. We fix the number of layers at 4 and experiment with different channel quantities, specifically at  $[u, 2u, 4u, 8u]$ , where  $u = [4, 8, 16, 32, 64]$ . The results indicate a gradual increase in accuracy with an increase in the number of channels, but the parameter count rises sharply. In the experimental plots, we observe that around  $u = 16$ ,

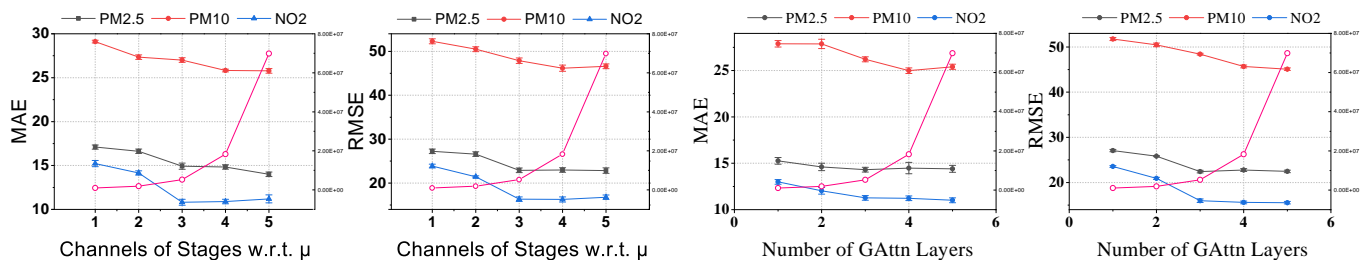


Fig. 11: Hyperparameter Study. In the scenario where the number of channels is given by  $[u, 2u, 4u, 8u]$ , with  $u$  taking values from the set  $[4, 8, 16, 32, 64]$ , the corresponding model performance is compared across various measurement metrics. Number of GATtn Layers represents the count of gradient-guided attention mechanism layers selected in our model. The magenta line represents the overall number of parameters in the model.

the performance of the experiment stabilizes. Further increasing the number of channels has little impact on the results but significantly increases the training parameters. Consequently, for the final experiment, we opt for  $u = 16$  as our channel parameter.

4) Spatial Sensitivity (Q4): Intra-city: The sensitivity of our model to extrapolated positions, especially at the edges, which are far from the city center, is a key consideration for tasks involving space extrapolation. The adaptability and generalization of the model to out-of-distribution content are crucial. Categorizing air quality monitoring stations into three groups based on geographical distribution, using z-scores. z-scores measure how far a data point is from the mean in terms of standard deviations. The formula is  $Z = (X - \mu)/\sigma$  where  $X$  is the data point,  $\mu$  and  $\sigma$  represent the mean and standard deviation of the data points, respectively.

We choose inference positions as shown in Fig. 10, particularly focusing on the edge positions. Baseline method is STGNP. Each position is marked with different colors and represented accordingly on the bar chart. Randomly chosen positions are represented in yellow on the bar chart. Context nodes are chosen randomly from the same test nodes, excluding the inference positions at this time. In experiments on the Beijing dataset, we find that the fully connected topological structure ensures adequate adjacency information, impacting regions less on extrapolation results. This underscores the robust distribution-capturing ability of the diffusion probability model, even in edge positions.

Cross-city: To validate the generalization capability of our proposed spatiotemporal extrapolation method, we conduct cross-city air quality experiments. Specifically, we treat the London air quality dataset as the target domain for cross-domain extrapolation, selecting the current state-of-the-art extrapolation method STGNP and the diffusion-based method CaPaint as comparison baselines. All models are trained on the Beijing air quality dataset and then evaluated on the London dataset for predicting PM2.5, PM10, and NO2 concentrations.

The experimental results demonstrate that our model achieves significant improvements over the comparison methods across all air quality indicators in cross-domain learning tasks. Particularly for the NO2 indicator, where

baseline methods exhibit high cross-domain inference errors on London data, our model effectively ensures cross-domain inference performance through diffusion mechanisms that generate representations consistent with the target distribution. This validates the effectiveness and generalization capability of our proposed method.

## V. RELATED WORK

### A. Spatiotemporal Extrapolation

The goal of spatiotemporal extrapolation tasks is to predict the state of never seen before spatiotemporal points by leveraging existing data and models through inference and prediction methods. Early approaches employed statistical and machine learning methods, such as kriging [34], these methods have limitations in complex scenarios due to underlying assumptions. KNN [27] and RF [28] are both computationally complex and exhibited sensitivity to outliers. Gaussian Processes [35] use flexible kernels to learn spatiotemporal dependencies. However, constructing kernels is computationally demanding. Matrix completion methods [11], capture spatiotemporal patterns with a low-rank matrix assumption. However, this approach become ineffective when confronting with scenarios that did not adhere to the low-rank assumption.

In neural network methods, Cheng et al. [30] propose a model, utilizing recurrent neural networks and multilayer perceptrons. Han et al. [36] improve performance by combining graph convolutional networks with a multi-channel attention module. Above methods are not considered uncertain and lack exploration of topological relationships. Wu et al. [5] proposed an inductive method, involving sampling different subgraphs and reconstructing them. However, it cannot leverage covariates information relevant to the target variables and fails to capture dynamic changes in node correlations.

Other work introduces generative models to learn the inherent distribution of spatiotemporal data and capture uncertainty. Hu et al. [2] employed neural processes, utilizing the KL divergence between the target node and context nodes as a guarantee for the correlation between nodes during training. Zhang et al. [37] considered different scales of geographical space in real-world scenarios. The former may face instability in training due to the simultaneous consideration of prediction accuracy



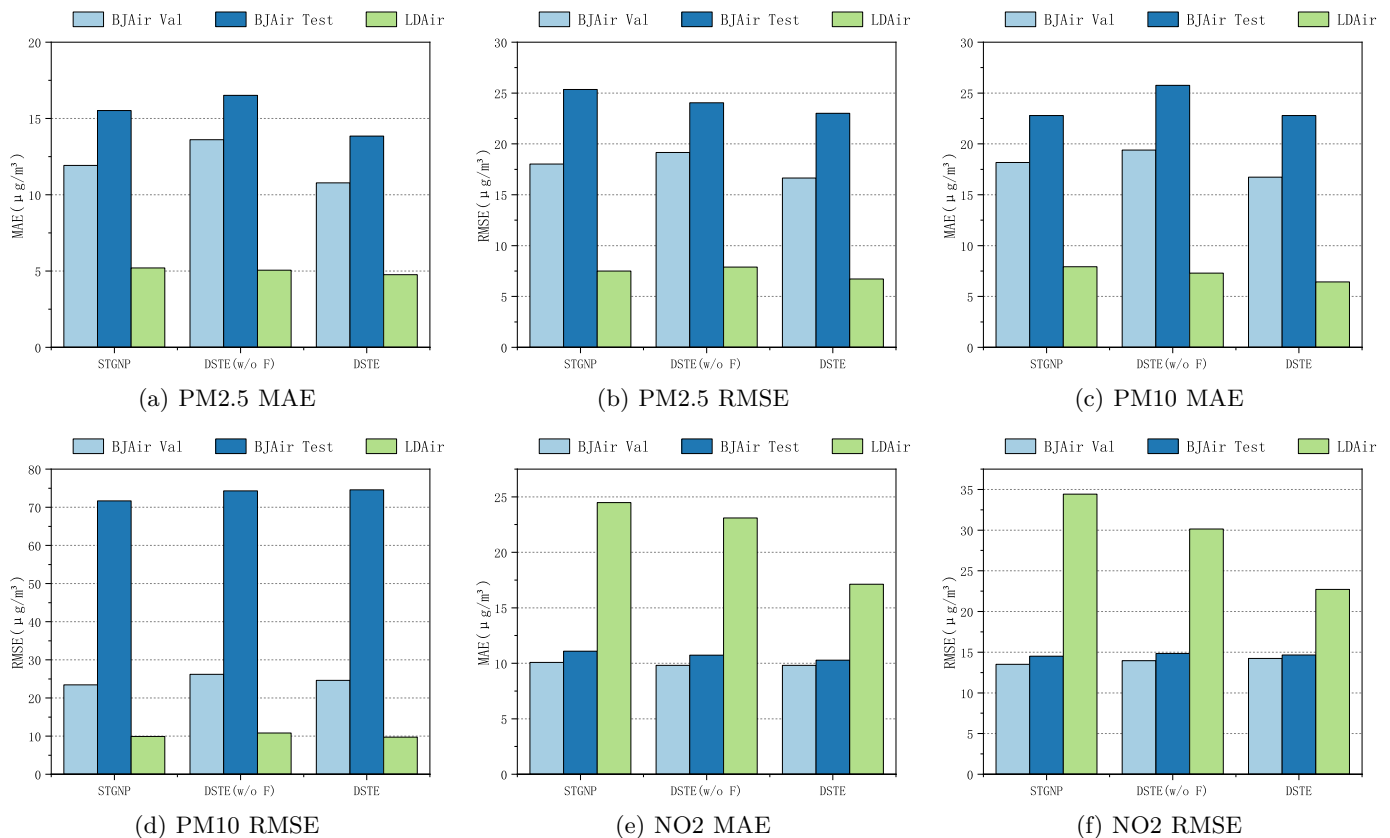


Fig. 12: Cross-city air quality prediction performance comparison. This figure presents cross-domain extrapolation results where models are trained on Beijing and tested on London. The comparison includes STGNP, DSTE(w/o F) (our method without non-local factor), and our full method across three air quality indicators (PM2.5, PM10, NO2) using MAE and RMSE metrics. (a)-(b) PM2.5 prediction results; (c)-(d) PM10 prediction results; (e)-(f) NO2 prediction results. Our method outperforms baselines across all metrics, with particularly significant improvements in NO2 prediction, validating the effectiveness and generalization capability of the proposed diffusion model in cross-domain scenarios.

and node neighborhood consistency, while the latter is struggling in training instability and pattern collapse [38].

## B. Spatiotemporal Diffusion Probability Model

The early proposal of the diffusion probability model was put forth by Sohl-Dickstein et al. [39] and later refined by Ho et al. [18], resulting in its application in the field of image generation, known as DDPM. Due to its notable performance and stability relative to other generative models [40], DDPM has garnered widespread attention.

Tashiro et al. [14] introduced the first diffusion probability model imputation framework for spatiotemporal data, named CSDI which utilizes self-attention mechanisms to independently learn temporal and feature-wise correlations. Building on this work, Liu et al. [3] considered spatial topology to enhance spatiotemporal correlations. However, the former struggles with capturing spatial correlations, making it challenging to address missing values in spatial contexts. The latter, spatial attention mechanism significantly increases the time complexity of model training and inference phase,  $n$  target points,  $m$  conditional information nodes,  $O((n+m)^2)$ , and  $n \ll m$ , while our

method achieves a time complexity of  $O(n^2)$ . Besides, our model is inductive, while this model is transductive.

Hu et al. [21] designed a universal spatiotemporal pre-training encoder to extract and compress conditional information. However, the shared spatiotemporal encoding module for both time and space completion limits the learning capability of the model in kriging tasks. This results in the model being able to only interpolate spatiotemporal data for positions that have been previously trained, lacking the ability to infer positions that have never been encountered, as well as not being able to perform the spatiotemporal extrapolation tasks mentioned in our paper. Zheng et al. [6] propose a diffusion model-based image super-resolution technique that extracts fine-grained information from coarse-grained data in urban settings. It is limited by the specific scenarios of super-resolution, posing challenges in effectively inferring dynamic and irregular spatiotemporal graph data. CaPaint [33] is a causal structure plugin for spatio-temporal prediction that leverages self-supervised reconstruction with a Vision Transformer to automatically identify causal and non-causal regions in the data, and employs a diffusion model to generatively inpaint non-causal regions, thereby improving the generalizability and



interpretability of the model, particularly in scenarios with scarce data or distribution shifts.

## VI. CONCLUSION

Aiming to predict values of never seen before region by utilizing information from neighboring nodes and exogenous covariates within the target region, we introduce conditional diffusion probability models, leveraging their robust capability to capture sequence distributions and conditional generative capacity. To explore the relationship between covariates and target variables and utilize them, we integrate a Non-local Factor Learning module to comprehensively combine information, and dynamic graph generation captures evolving topology. In extensive experiments, the proposed method demonstrates a significant improvement in accuracy compared to previous state-of-the-art approaches. Although our method performs well on datasets such as weather quality, it encounters inadequacies when dealing with tasks like traffic flow prediction. Traffic patterns are often influenced by unpredictable non-natural factors such as traffic accidents. How to extract the true physical or causal correlation between covariates and target variables in noise information will be worth exploring.

## ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (2022YFB3103700, 2022YFB3103702), the National Natural Science Foundation of China (62272193, 62472194), and the Major Science and Technology Project of Jilin Province (20240212002GX).

## References

- [1] En Wang, Weiting Liu, Wenbin Liu, Yongjian Yang, Bo Yang, and Jie Wu. Spatiotemporal urban inference and prediction in sparse mobile crowdsensing: A graph neural network approach. *IEEE Transactions on Mobile Computing*, 2022.
- [2] Junfeng Hu, Yuxuan Liang, Zhencheng Fan, Hongyang Chen, Yu Zheng, and Roger Zimmermann. Graph neural processes for spatio-temporal extrapolation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 752–763, New York, NY, USA, 2023. Association for Computing Machinery.
- [3] Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. Pristi: A conditional diffusion framework for spatiotemporal imputation. *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1927–1939, 2023.
- [4] Guodong Xu, Hai Wang, Shuo Ji, Yuhui Ma, and Yi Feng. Mpformer: A transformer-based model for earthen ruins climate prediction. *Tsinghua Science and Technology*, 29(6):1829–1838, 2024.
- [5] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4478–4485, 2021.
- [6] Yuhao Zheng, Lian Zhong, Senzhang Wang, Yu Yang, Weixi Gu, Junbo Zhang, and Jianxin Wang. Diffuflow: Robust fine-grained urban flow inference with denoising diffusion model. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3505–3513, 2023.
- [7] Mingtao Sun, Yan Wei, Shan Jiang, and Guozhu Jia. A comprehensive framework for predicting public opinion by tracking multi-informational dynamics. *Frontiers of Computer Science*, 18(4):184344, 2024.
- [8] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *IEEE communications Magazine*, 49(11):32–39, 2011.
- [9] Wenbin Liu, Yongjian Yang, En Wang, Hengzhi Wang, Ziheng Wang, and Jie Wu. Dynamic online user recruitment with (non-) submodular utility in mobile crowdsensing. *IEEE/ACM Transactions on Networking*, 29(5):2156–2169, 2021.
- [10] Daqing Zhang, Haoyi Xiong, Leye Wang, and Guanling Chen. Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, page 703–714, New York, NY, USA, 2014. Association for Computing Machinery.
- [11] En Wang, Mijia Zhang, Wenbin Liu, Haoyi Xiong, Bo Yang, Yongjian Yang, and Jie Wu. Outlier-concerned data completion exploiting intra- and inter-data correlations in sparse crowdsensing. *IEEE/ACM Transactions on Networking*, 2022.
- [12] En Wang, Weiting Liu, Wenbin Liu, Yongjian Yang, Bo Yang, and Jie Wu. Spatiotemporal urban inference and prediction in sparse mobile crowdsensing: A graph neural network approach. *IEEE Transactions on Mobile Computing*, 22(11):6784–6799, 2023.
- [13] Xu Wang, Hongbo Zhang, Pengkun Wang, Yudong Zhang, Binwu Wang, Zhengyang Zhou, and Yang Wang. An observed value consistent diffusion model for imputing missing values in multivariate time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2409–2418, 2023.
- [14] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.
- [15] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.
- [16] Yunhao Liu, Kebin Liu, and Mo Li. Passive diagnosis for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 18(4):1132–1144, 2010.
- [17] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [19] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.
- [20] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International conference on machine learning*, pages 1704–1713. PMLR, 2018.
- [21] Junfeng Hu, Xu Liu, Zhencheng Fan, Yuxuan Liang, and Roger Zimmermann. Towards unifying diffusion models for probabilistic spatio-temporal graph learning. *arXiv preprint arXiv:2310.17360*, 2023.
- [22] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3575–3584, 2019.
- [23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [24] Kdd-cup18. [https://www.biendata.xyz/competition/kdd\\_2018/data/](https://www.biendata.xyz/competition/kdd_2018/data/).
- [25] Intellab. <https://db.csail.mit.edu/labdata/labdata.html>.

- [26] George Y Lu and David W Wong. An adaptive inverse-distance weighting spatial interpolation technique. *Computers & geosciences*, 34(9):1044–1055, 2008.
- [27] Pedregosa Fabian. Scikit-learn: Machine learning in python. *Journal of machine learning research* 12, page 2825, 2011.
- [28] Matthias Schonlau and Rosie Yuyan Zou. The random forest algorithm for statistical learning. *The Stata Journal*, 20(1):3–29, 2020.
- [29] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [30] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. A neural attention model for urban air quality inference: Learning the weights of monitoring stations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [32] Kun Zheng, Qiya Tan, Huihua Ruan, Jinbiao Zhang, Cong Luo, Siyu Tang, Yunlei Yi, Yugang Tian, and Jianmei Cheng. Gan-argcprednet v2. 0: a radar echo extrapolation model based on spatiotemporal process enhancement. *Geoscientific Model Development*, 17(1):399–413, 2024.
- [33] Yifan Duan, Jian Zhao, Junyuan Mao, Hao Wu, Jingyu Xu, Caoyuan Ma, Kai Wang, Kun Wang, Xuelong Li, et al. Causal deciphering and inpainting in spatio-temporal dynamics via diffusion model. *Advances in Neural Information Processing Systems*, 37:107604–107632, 2024.
- [34] Margaret A Oliver and Richard Webster. Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information System*, 4(3):313–332, 1990.
- [35] Stephen Roberts, Michael Osborne, Mark Ebdon, Steven Reece, Neale Gibson, and Suzanne Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550, 2013.
- [36] Qilong Han, Dan Lu, and Rui Chen. Fine-grained air quality inference via multi-channel attention model. In *IJCAI*, pages 2512–2518, 2021.
- [37] Xiyuan Zhang, Ranak Roy Chowdhury, Jingbo Shang, Rajesh Gupta, and Dezhi Hong. Esc-gan: Extending spatial coverage of physical sensors. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1347–1356, 2022.
- [38] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE, 2020.
- [39] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- [40] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.



En Wang (Member, IEEE) received his B.E. degree in Software Engineering from Jilin University, Changchun, China, in 2011; and his M.E. and Ph.D. degrees in Computer Science and Technology from Jilin University, Changchun, China, in 2013 and 2016, respectively. He was also a joint Ph.D. student with the Department of Computer and Information Science, Temple University, Philadelphia, PA, USA. He is currently a professor in the College of Computer Science and Technology and the



vice dean of College of Software, Jilin University, Changchun, China. His current research focuses on mobile computing, crowd intelligence, and data mining.

Qinglun Meng received the bachelor's degree in chemistry from Jilin University, Changchun, China, in 2022. Currently, he is working toward the master's degree in computer science and technology with Jilin University, Changchun, China. His current research focuses on mobile crowdsensing, spatiotemporal data processing.



Wenbin Liu received the B.S. degree in physics and the Ph.D. degree in computer science and technology from Jilin University, China, in 2012 and 2020, where he is currently an associate professor with the College of Computer Science and Technology. He is currently also a postdoctoral researcher at China Telecom. His research interests include mobile crowdsensing, spatio-temporal crowdsourcing, and ubiquitous computing.



Bo Yang is currently a professor in the College of Computer Science and Technology, Jilin University. He is also the director of the Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, China. His current research interests are in the areas of data mining, complex network analysis, self-organized and self-adaptive multi-agent systems, with applications to knowledge engineering and intelligent health informatics.



Jie Wu (Fellow, IEEE) is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu was an IEEE Computer Society Distinguished Visitor and ACM Distinguished Speaker. He is a China Computer Federation (CCF) Distinguished Speaker and a Fellow of the IEEE. Currently, he is on leave working as a Scientist at China Telecom.