BiST: A Lightweight and Efficient Bi-directional Model for Spatiotemporal Prediction

Jiaming Ma
University of Science and Technology of China
Hefei, China
JiamingMa@mail.ustc.edu.cn

Binwu Wang
University of Science and Technology of China
Hefei, China
wbw1995@mail.ustc.edu

Pengkun Wang
University of Science and Technology of China
Hefei, China
pengkun@ustc.edu.cn

Zhengyang Zhou
University of Science and Technology of China
Hefei, China
zzy0929@ustc.edu.cn

Xu Wang
University of Science and Technology of China
Suzhou, China
wx309@ustc.edu.cn

Yang Wang
University of Science and Technology of China
Hefei, China
angyan@ustc.edu.cn

ABSTRACT

While existing spatiotemporal prediction models have shown promising performance, they often rely on the assumption of input-label spatiotemporal consistency, and their high complexity raises concerns about scalability. To enhance both efficiency and performance, we integrate label information into the learning process and propose a spatiotemporal dynamic theory that outlines a bi-directional learning paradigm. Building on this paradigm, we design BiST, a lightweight yet effective **Bi**-directional **S**patio-**T**emporal prediction model. BiST incorporates two key processes: a forward spatiotemporal learning process and a backward correction process. The forward process utilizes MLP layers exclusively to model input correlations and generate base prediction. In the backward process, we implement a spatiotemporal decoupling module, which can learn the residual modeling deviation between input and label representations from a decoupled perspective. After smoothing the residual with a diffusion module, we can obtain the correction term to correct the base predictions. This innovative design enables BiST to achieve competitive performance while remaining lightweight. We evaluate BiST against 26 baselines across 13 datasets, including a large-scale dataset with ten thousand nodes and a long-range dataset spanning 20 years. An impressive experimental result demonstrates that BiST achieves a 8.13% improvement in performance compared to state-of-the-art models while consuming only 1.86% of the training time and 7.36% of the memory usage.

PVLDB Reference Format:
Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. BiST: A Lightweight and Efficient Bi-directional Model for Spatiotemporal Prediction. PVLDB, 18(6): 1663 - 1676, 2025.
doi:10.14778/3725688.3725697

PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at https://github.com/PoorOtterBob/BiST.

1 INTRODUCTION

With significant advancements in GPS technology and sensor monitoring devices, researchers have amassed extensive urban data, characterized by both temporal and spatial attributes, collectively referred to as spatiotemporal data. This wealth of spatiotemporal data has fueled the growth of urban computing. Within this domain, spatiotemporal prediction, a fundamental task, has garnered considerable attention from both industry and academia. This task aims to leverage historically observed spatiotemporal data to forecast future values [30, 40, 73, 74, 77].

In the field of spatiotemporal prediction, the popular tool is spatiotemporal graph convolutional networks, which consist of different temporal and spatial modules for capturing temporal and spatial correlations respectively. To improve prediction performance, researchers have focused on enhancing the representation capabilities of these modules through various advanced techniques. Currently, Transformer-based models dominate the spatiotemporal prediction task, such as D^{2}STGNN [49] and STAEformer [32]. Despite their encouraging success, there remain two limitations.

Input-label spatiotemporal deviation. Existing models typically employ a forward spatiotemporal learning process that captures spatiotemporal correlations from input data, generates label representations, and uses these label representations for prediction. This implicitly assumes consistency between the spatiotemporal

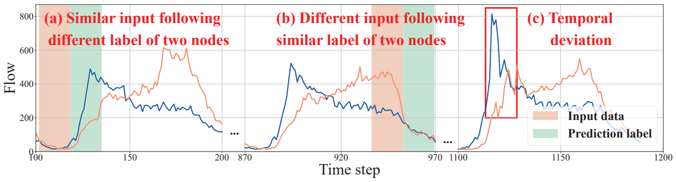


Figure 1: Three cases of spatiotemporal deviation in the spatial and temporal dimensions. (a) and (b) demonstrate the spatial deviation across pairs of nodes. (c) illustrates the temporal deviation within a single node.

Yang Wang and Binwu Wang are the corresponding authors.

Proceedings of the VLDB Endowment, Vol. 18, No. 6 ISSN 2150-8097.
doi:10.14778/3725688.3725697

correlations in the input data and those in the labels. However, this assumption is overly idealistic; spatiotemporal correlations between input and labels may exhibit significant differences in both spatial and temporal dimensions, which we define as spatiotemporal deviation. We illustrate this concept using the Large-SD dataset [33] as an example. Figure 1 (a) and (b) demonstrate the spatial deviation. Specifically, similar input following different label of Figure 1 (a): while two nodes have similar input data distributions, their subsequent label similarities differ significantly. Different input following similar label of Figure 1 (b): despite two nodes having significantly different input data, their labels exhibit similar distributions. These node pairs are indistinguishable to the model, as the model tends to make similar (different) predictions for nodes with similar (different) inputs, thereby reducing prediction accuracy. In the temporal dimension, spatiotemporal deviation manifests as sudden increases or decreases of the data, as shown in Figure 1 (c). Although several studies have proposed potential solutions to tackle spatiotemporal deviations using node embedding techniques [48] or by extending input sequence lengths [10], we contend that the limited utilization of label information continues to hinder these models in effectively addressing the spatiotemporal deviation problem.

Expensive computational complexity. While existing models achieve performance improvements, they also increase time and memory complexities. Regarding time complexity, transformer-based spatiotemporal layers exhibit quadratic growth as the number of nodes increases [46, 49]. In terms of memory occupancy, these models often stack multiple complex spatiotemporal layers to enhance their representational capabilities. Since their loss function for regression tasks relies on the computational gradient graphs of all nodes for backpropagation, the GPU must maintain a gradient matrix for nodes at each layer, leading to significant memory overhead. The heavy computational burden limits the scalability of these models on large-scale spatiotemporal data.

In this paper, we aim to advance both efficiency and performance. Regarding performance, we break from the spatiotemporal consistency assumption between input and labels followed by existing models, explicitly incorporating label information during training to better model spatiotemporal deviations. This design allows us to deviate from the trend of stacking multiple spatiotemporal layers, opting instead for lightweight MLP as the backbone. Ultimately, the proposed model achieves competitive predictive performance while maintaining high training efficiency and low memory utilization, as illustrated in Figure 2.

Specifically, we propose a spatiotemporal dynamics theory that guides a rational prediction process by incorporating label information. This theory reveals that the final prediction should be influenced by two components: a base prediction, derived from modeling spatiotemporal correlations of the input data, and a correction term, generated by modeling the residuals that represent spatiotemporal deviation between labels and input. Based on this theory, we propose the **Bi**-directional **S**patio-**T**emporal prediction model (BiST), which includes a forward spatiotemporal learning process and a backward residual correction process. In the forward process, we only use MLP layers to capture time dependencies at different granularities, generating the base prediction. To model residuals accurately in the backward process, we introduce a spatiotemporal decoupling residual learning module that separates spatiotemporal features into node-shared context features and node-personalized features, effectively capturing spatiotemporal deviation between label and input representations. After smoothing the residuals with a diffusion operator, we decode the residuals to generate correction terms, correcting the predictions. Note that our model utilizes high-dimensional label representations rather than actual labels for residual modeling, enhancing its capability to represent residuals. During the training process, the model can effectively learn the spatiotemporal deviation, which will be beneficial for the inference phase. Evaluated on 13 spatiotemporal datasets, our model demonstrates competitive performance while maintaining computational efficiency and low memory overhead. Our contributions are summarized as follows:

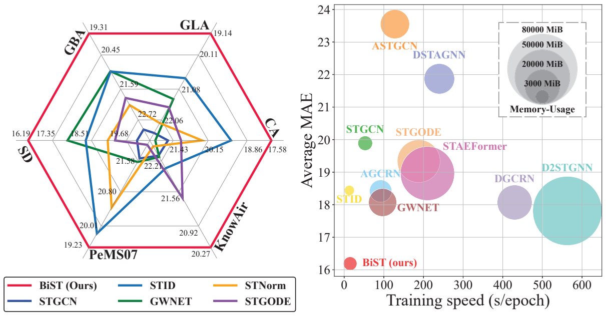


Figure 2: Model performance and efficiency comparison. The left figure illustrates the comparative prediction performance of various models on six datasets. The right figure showcases the training time per epoch and memory usage of each model on LargeST-SD 2019 dataset.

- We develop a spatiotemporal dynamic theory that establishes a novel bi-directional spatiotemporal learning paradigm incorporating label modeling.
- Based on this theory, we design a lightweight and effective bidirectional spatiotemporal prediction model, which includes a forward spatiotemporal learning process and a backward residual correction process.
- We introduce a spatiotemporal residual learning module that models the spatiotemporal deviation features between input and label from the decoupling perspective.
- Extensive experiments with 26 models on 13 datasets have demonstrated that our model achieves competitive performance, maintaining high efficiency and low memory usage.

2 RELATED WORK

Time series prediction. In recent years, multivariate time series forecasting has garnered significant attention due to its wide-ranging applications in fields such as finance, healthcare, and environmental monitoring [19, 23, 35, 43, 53, 55, 69]. Among the various approaches, Transformer [51] has emerged as a prominent framework, achieving remarkable success in sequence prediction tasks. However, the inherent high time complexity of the Transformer architecture has driven researchers to explore more efficient and innovative methods. For instance, Preformer [6] and PatchTST [42] have introduced patch-based strategies to improve computational efficiency. Meanwhile, Reformer [6] has incorporated locality-sensitive hashing to enhance the self-attention mechanism. Another notable category of approaches leverages lightweight MLPs as their backbone [59, 71]. For example, TimeMixer [61] integrates multi-scale temporal decoupling to boost MLP performance, while SOFTS [17] employs a centralized strategy to model dependencies across channels. Despite these advancements, these models primarily focus on capturing temporal dependencies and often overlook the spatial dependencies inherent in spatiotemporal data. As a result, their performance generally falls short compared to state-of-the-art spatiotemporal forecasting models.

Spatiotemporal prediction. Spatiotemporal prediction task which aims to use past observations to predict future values is fundamental to smart city applications [26, 57, 58, 76]. With the remarkable success of GCN in various fields [14, 15, 52, 78], the current trends of this field revolves around designing cutting-edge spatiotemporal graph convolutional networks [54, 72]. For example, DCRNN [29], introduced a novel diffusion convolution that works in conjunction with GRU. STGCN [67] have replaced RNN with extended causal convolutions for time pattern modeling. With the rise of Transformers in the natural language and visual domains, the latest trend is shifting towards the use of Transformers and their variants for spatiotemporal prediction [7, 21]. For example, D^{2}STGNN [49] and STAEformer [32] use self-attention mechanisms from Transformers for dynamic graph learning, combined with proposed spatiotemporal embedding techniques.

Although these models significantly improve predictive performance, they also pose challenges due to their substantial computational complexity and memory overhead. Additionally, these models adhere to the input-label consistency assumption, which limits their ability to effectively handle inconsistent information.

3 PROBLEM DEFINITION

Spatiotemporal data. Spatiotemporal data are represented as a multivariate time series comprising multiple time-dependent variables, such as observations collected from sensors. We formulate the multivariate time series from the time step m to the time step n as a tensor $X_{m:n} \in \mathbb{R}^{(n-m+1)\times N\times c}$, where N denotes the number of variables (e.g. sensors) and c indicates the number of channels.

Spatiotemporal graph. Each variable depends not only on its past values, but also on other variables. Such dependencies are captured by a spatiotemporal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V}$ is a set of $|V| = N$ nodes, each node corresponding to a sensor or air quality monitor. The set of edges is denoted by $\mathcal{E}$, and $\mathbf{A} \in \mathbb{R}^{N\times N}$ represents the adjacency matrix, which can be modeled using a predefined metric, such as the distance between nodes, or can be adaptively learned from the data end-to-end.

Spatiotemporal prediction. Given the observed multivariate time series $X_{t-T+1:t} \in \mathbb{R}^{T\times N\times c}$ from the previous T time steps, the goal is to learn a function f to forecast spatiotemporal data for the next T_P time steps. This mapping can be formally defined as:

$$\hat{\mathrm{X}}_{t+1:t+T_p} = f(X_{t-T+1:t}) \in \mathbb{R}^{T_P\times N\times c}. \tag{1}$$

Table 1: Some important notations description.

Notations	Description
$\mathbf{X}, X$	Input data and its random variable in GMRF.
Y, Y	Label data and its random variable in GMRF.
$\mathrm{Y}_{\text{base}}/\mathrm{Y}_{\text{cor}}$	Base/Correction prediction.
$\mathbf{Z}_{\text{In}}/\mathbf{Z}_{\text{La}}/\mathbf{Z}_{\text{R}}$	Input/Label/Residual representation.
$\boldsymbol{e}_P, \boldsymbol{e}_T, \boldsymbol{e}_D, \boldsymbol{e}_S$	Spatiotemporal prompt embedding.
E_q/E_k	Personalized-feature /Context-feature embedding.

4 METHODOLOGY

We develop a spatiotemporal dynamics theory to establish a rational paradigm for spatiotemporal prediction. Building on this theory, we design the spatiotemporal prediction model BiST, and subsequently provide a detailed explanation of each component within BiST. For clear presentation, we use $\mathbf{X}$ (i.e., $X_{t-T+1:t}$) and Y to represent the input and the corresponding label, respectively.

4.1 Spatiotemporal Dynamics Theory

Gaussian Markov Random Field (GMRF) is a widely used tool for modeling complex dependencies among random variables in a structured manner, particularly in spatiotemporal dynamic analysis [9, 75]. In line with these studies, we also employ a GMRF model to represent spatiotemporal data, where each spatiotemporal data point is associated with a variable in the GMRF. Subsequently, we analyze the dependencies between these variables. In the following sections, spatiotemporal data points will be represented using regular font, while their corresponding random variables in the GMRF will be denoted in *italic font.*

Let's consider the corresponding variable of node u at future time step t, which is denoted as $Y_{t,u} \in \mathbb{R}^c$, there are correlations between $Y_{t,u}$ and the variable of the other nodes[1], which is denoted as $Y_{t,\hat{u}} \coloneqq \left[Y_{t,1}^\top, \ldots, Y_{t,u-1}^\top, Y_{t,u+1}^\top, \ldots, Y_N^\top\right]^\top \in \mathbb{R}^{(N-1)\times c}$. We incorporate this correlation into the GMRF model.

Theorem 1. If we integrate the label information into GMRF, we can use it as a condition of the GMRF to predict the value $\hat{\mathrm{Y}}_{t,u}$ of variable $Y_{t,u}$ with the aim of minimizing the difference from the label $\mathrm{Y}_{t,u}$. For any future time step $t = \{1, 2, ..., T_P\}$, the expectation of $Y_{t,u}$ with respect to $\mathbf{X}$ and $\mathrm{Y}_{t,\hat{u}}$ is

$$\mathbb{E}\left[Y_{t,u}|\mathbf{X}, \mathrm{Y}_{t,\hat{u}}\right] = \underbrace{\mathbb{E}\left[Y_{t,u}|\mathbf{X}\right]}_{\text{Base prediction}} + \underbrace{\underbrace{\beta_{t,u}\left(\mathbf{I}_N + \alpha_t \mathcal{A}\left(\mathbf{A}\right)\right)_{u,\hat{u}}}_{\text{Diffusion Kernel}} \times_2 \underbrace{\mathbf{c}_{t,\hat{u}}}_{\text{Residual}}}_{\text{Correction}}. \tag{2}$$

This equation indicates that, when incorporating label information, the spatiotemporal prediction paradigm should consist of a **base prediction** and a **correction term**. The detailed proof of this proposition is provided in Section A.

The correction term consists of two elements: the diffusion kernel and the residual. The $\beta_{t,u}$ is a scalar coefficient calculated by:

$$\beta_{t,u} = \left[\left(1 + \alpha_t\right)\left(1 + \alpha_t \mathcal{A}\left(\mathbf{A}\right)_{u,u}\right)\right]^{-1}, \tag{3}$$

---

[1] To reduce the algorithm complexity, we focus solely on the spatial correlation at each time step when modeling label features. We validate it in the experimental section 5.8.

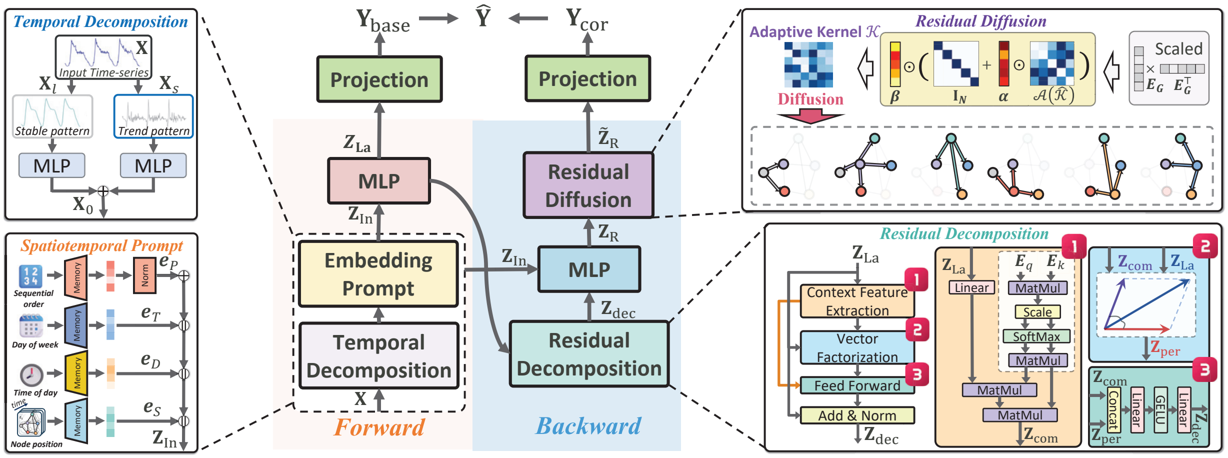


Figure 3: Details of the proposed model BiST. Our framework comprises a forward spatiotemporal learning process and a backward residual correction process.

where α_t is a scalar that controls the strength of residual propagation, and $\mathcal{A}(\mathbf{A})_{u,u}$ indicates the entry on u-th row and u-th column of $\mathcal{A}(\mathbf{A})$. $\mathcal{A}(\mathbf{A}) = \mathbf{I}_N - \tilde{\mathbf{A}}$ where $\mathbf{I}_N$ is the identity matrix of the adjacency matrix $\mathbf{A}$, and $\tilde{\mathbf{A}}$ is a normalization version of $\mathbf{A}$. $(\mathbf{I}_N + \alpha_t \mathcal{A}(\mathbf{A}))_{u,\hat{u}} \in \mathbb{R}^{1\times(N-1)}$ is the u-th row of $\mathbf{I}_N + \alpha_t \mathcal{A}(\mathbf{A})$ exclude itself. The residual term $\mathbf{c}_{t,\hat{u}}$ represents the difference between base prediction and label expectations:

$$\mathbf{c}_{t,\hat{u}} \coloneqq \mathbb{E}\left[Y_{t,\hat{u}}|\mathbf{X}\right] - \mathbb{E}\left[Y_{t,\hat{u}}\right] \in \mathbb{R}^{1\times(N-1)\times c}. \tag{4}$$

In fact, the base prediction is derived from modeling the correlations within the input data, while the label distribution is influenced by the autocorrelation of the labels. Consequently, the residual term captures the discrepancies in features between the input and label. **Summary**. Our theory indicates that a prediction paradigm incorporating label information should consist of two components: a forward spatiotemporal learning process generating base predictions and a residual correction process modeling spatiotemporal residuals to correct predictions. Although existing models integrate various techniques to enhance the former process, they do not explicitly include a correction process that utilizes label features.

4.2 Overview of the proposed BiST

Based on the proposed theoretical, we develop BiST, designed to incorporate a forward spatiotemporal learning process and a backward residual correction learning process. This structure is illustrated in Figure 3 and is detailed in Algorithm 1.

Forward spatiotemporal learning consists of a spatiotemporal learning layer and a spatiotemporal embedding prompt layer. The former layer integrates the knowledge of the time structure and decomposes the time series $\mathbf{X}$ into stable components and trend components, which can reduce the model's learning complexity. The spatiotemporal embedding prompt encodes prior knowledge to help the model achieve comprehensive spatiotemporal learning. Through these two layers, we can obtain the input representation $\mathbf{Z}_{\text{In}}$. Finally, $\mathbf{Z}_{\text{In}}$ is fed into the MLP layers for spatiotemporal learning and outputs label representations $\mathbf{Z}_{\text{La}}$, which are then inputted into a predictor to generate base predictions $\mathbf{Y}_{\text{base}}$.

Backward residual correction consists of a residual decouple layer and a residual diffusion layer. The former layer is used to model inconsistencies between input representations $\mathbf{Z}_{\text{In}}$ and label representations $\mathbf{Z}_{\text{La}}$, i.e., residual term $\mathbf{Z}_{\text{R}}$. Then a residual diffusion layer uses the affinity between nodes to smooth the residual term. Finally, the output is fed into a fully connected layer to generate correction predictions $\mathbf{Y}_{\text{corr}}$ used to correct the base predictions $\mathbf{Y}_{\text{base}}$ to generate a more accurate prediction $\mathbf{Y}$.

4.3 Forward Spatiotemporal Learning

4.3.1 Temporal decomposition. In the time series community, researchers [63, 71] decompose time series into components with different time granularities. Inspired by these works, we also adopt a temporal decomposition layer, which uses the padding moving average operation $\text{AvgPool}(\cdot; k)$ with kernel size k to decouple the input $\mathbf{X} \in \mathbb{R}^{T\times N\times c}$ into stable patterns $\mathbf{X}_l$ and trend patterns $\mathbf{X}_s$:

$$\mathbf{X}_l = \text{AvgPool}(\mathbf{X}; k) \in \mathbb{R}^{T\times N\times c}, \tag{5}$$

$$\mathbf{X}_s = \mathbf{X} - \mathbf{X}_l \in \mathbb{R}^{T\times N\times c}. \tag{6}$$

Concretely, we pad the data along the temporal dimension in AvgPool $(\cdot; k)$ to keep the corresponding series length after pooling. Then, we use two MLP layers to capture the spatiotemporal dependencies of these two components, respectively. Then, we splice the outputs to generate the final output $\mathbf{X}_0$.

$$\mathbf{X}_0 = \text{MLP}_1(\mathbf{X}_l) + \text{MLP}_2(\mathbf{X}_s) \in \mathbb{R}^{T\times N\times d_s}, \tag{7}$$

4.3.2 Spatiotemporal embedding prompt. Spatiotemporal prompt learning aims to utilize various additional information to prompt models to learn more comprehensive spatiotemporal patterns. Drawing inspiration from existing work [70], we introduce spatiotemporal embedding techniques to encode spatiotemporal prior information (such as timestep-of-day and day-of-week information) and integrate these beneficial embeddings into the model, thereby enhancing its learning capabilities through prompting.

In the temporal dimension, we design a temporal embedding including two embedding vectors: timestamp-of-day embedding and day-of-week embedding to capture periodic temporal dependencies. The first embedding, $\boldsymbol{e}_T \in \mathbb{R}^{N_T \times d_P}$, encodes the time-step position information of a day, where N_T represents the number of sampling points in a day. For instance, in the PeMS system, with a data sampling frequency of five minutes, N_T is equal to 288. The second embedding, $\boldsymbol{e}_D \in \mathbb{R}^{N_D \times d_P}$, encodes the positions of different days of the week. Here, $N_D = 7$ corresponds to the number of days in a week and d_P denotes the dimension of the representations. In the spatial dimension, we employ an adaptive node-level embedding, $\boldsymbol{e}_S \in \mathbb{R}^{N \times d_P}$, to account for the heterogeneous data distributions between the N nodes. These parameterized embeddings are updated end-to-end with the model.

In addition, following Transformer [51], we also integrate the sequential information of each input data point into the input $\mathbf{X}_0$. Finally, we integrate the temporal and spatial embedding into the model, and the output $\mathbf{Z}_{\text{In}}$ is denoted as the **input representation:**

$$\mathbf{Z}_{\text{In}} = [\mathbf{X}_0 \| \boldsymbol{e}_T \| \boldsymbol{e}_D \| \boldsymbol{e}_S] \in \mathbb{R}^{T \times N \times (d_s + 3 * d_P)}. \tag{8}$$

4.3.3 Forward prediction. We employ L MLP layers for spatiotemporal forward learning to effectively capture the spatiotemporal features of the input data. For the input to the l-th layer, denoted as $\mathbf{Z}_{\text{f}}^{(l)} \in \mathbb{R}^{T \times N \times d^{(l)}}$, we start with $\mathbf{Z}_{\text{f}}^{(0)} = \mathbf{Z}_{\text{In}}$. The forward process of this MLP layer is defined as follows:

$$\mathbf{Z}_{\text{f}}^{(l+1)} = \text{GELU}\left(\mathbf{Z}_{\text{f}}^{(l)} W_1^{(l)} + b_1^{(l)}\right) W_2^{(l)} + b_2^{(l)} + \mathbf{Z}_{\text{f}}^{(l)}, \tag{9}$$

where $l \in \{0, 1, ..., L-1\}$ and GELU $(\cdot)$ is activation function. $W_1^{(l)} \in \mathbb{R}^{d^{(l)} \times 4d^{(l)}}$, $W_2^{(l)} \in \mathbb{R}^{4d^{(l)} \times d^{(l+1)}}$ and biases $b_1^{(l)} \in \mathbb{R}^{4d^{(l)}}$, $b_2^{(l+1)} \in \mathbb{R}^{d^{(l+1)}}$ are learnable parameters. The final output, $\mathbf{Z}_{\text{La}} = \mathbf{Z}_{\text{f}}^{(L)} \in \mathbb{R}^{T \times N \times d_{\text{hid}}}$, is denoted as **label representation**. Finally, we use a MLP layer as decoder to generate base prediction:

$$\mathbf{Y}_{\text{base}} = \mathbf{Z}_{\text{La}} W_{\text{out}} + b_{\text{out}} \in \mathbb{R}^{T_P \times N \times c}, \tag{10}$$

where $W_{\text{out}} \in \mathbb{R}^{(T \times d_{\text{out}}) \times (T_P \times c)}$ and $b_{\text{out}} \in \mathbb{R}^{T_P \times c}$ are learnable.

4.4 Backward Residual Correction

The backward residual correction process learns spatiotemporal deviation features of the input label, i.e., the residual term, to generate correction predictions. This process comprises two modules: a residual learning module and a residual diffusion module.

4.4.1 Spatiotemporal residual learning. To model the residual terms between the labels and the inputs, we use label representations generated from the forward process instead of directly using the labels since labels are unavailable during inference in the inference phase. More importantly, label representations, which are high-dimensional features of labels [28, 45], contain rich information that allows the model to learn residual terms flexibly.

For residual learning, we design a residual decoupling module that decomposes spatiotemporal features into contextual and personalized features. The contextual features, influenced by environmental attributes, may be shared among nodes. In contrast, the latter are affected by mutation factors (such as temporary traffic control at specific intersections), leading to inconsistencies between node inputs and label features.

Specifically, we first initialize two parameterized embeddings using a normal random distribution: $E_k \in \mathbb{R}^{K \times d}$ to capture contextual features with K virtual clusters and $E_q \in \mathbb{R}^{N \times d}$ to learn fine-grained node-specific features. Parameterized embeddings adaptively capture high-level features as the model undergoes end-to-end updates. Then, we compute the receptive coefficient between nodes and virtual clusters as follows:

$$S = \frac{E_q E_k^{\top}}{\sqrt{d}} \in \mathbb{R}^{N \times K}. \tag{11}$$

where S records the affinity between each node and the virtual clusters. Then we calculate the similarity between the nodes according to the macroscopic features: $\mathcal{W}\left(E_q, E_k\right) = \overline{S} \times \overline{S}^{\top} \in [0,1]^{N \times N}$. Here, $\overline{S}$ is the normalization version of S by softmax operation. Finally, given current label representation $\mathbf{Z}_{\text{La}}$, we can aggregate the information of the neighborhood nodes and extract the context feature representation, which is denoted as $\mathbf{Z}_{\text{com}}$:

$$E_v = \mathbf{Z}_{\text{La}} W_v + b_v \in \mathbb{R}^{T \times N \times d_{\text{out}}}, \tag{12}$$

$$\mathbf{Z}_{\text{com}} = \mathcal{W}\left(E_q, E_k\right) \times_2 E_v \in \mathbb{R}^{T \times N \times d_{\text{out}}}, \tag{13}$$

where $\times_2$ means matrix multiplication in the node dimension. Finally, we obtain personalized feature representation $\mathbf{Z}_{\text{per}}$ as follows:

$$\mathbf{Z}_{\text{per}} = \mathbf{Z}_{\text{La}} - \mathbf{Z}_{\text{com}} \in \mathbb{R}^{T \times N \times d_{\text{out}}}. \tag{14}$$

To model the difference between input representation and label representation, we first align $\mathbf{Z}_{\text{per}}$ and $\mathbf{Z}_{\text{com}}$ with input representation $\mathbf{Z}_{\text{In}}$ in their channel dimensions:

$$\mathbf{Z}_{\text{dec}} = \text{GELU}\left(\left[\mathbf{Z}_{\text{per}} \| \mathbf{Z}_{\text{com}}\right] W_1 + b_1\right) W_2 + b_2 \in \mathbb{R}^{T \times N \times d_{\text{hid}}}. \tag{15}$$

Then we calculate the inconsistency information between two representations: $\mathbf{Z}_{\text{b}}^{(0)} = \mathbf{Z}_{\text{In}} - \mathbf{Z}_{\text{dec}}$, then we use MLP with L layers to capture high-dimensional features, and the final output is denoted as the residual representation $\mathbf{Z}_{\text{R}} = \mathbf{Z}_{\text{b}}^{(L)}$. The forward process of each MLP layer is as follows:

$$\mathbf{Z}_{\text{b}}^{(l+1)} = \text{GELU}\left(\mathbf{Z}_{\text{b}}^{(l)} W_3^{(l)} + b_3^{(l)}\right) W_4^{(l)} + b_4^{(l)} + \mathbf{Z}_{\text{b}}^{(l)}, \tag{16}$$

where $l \in \{0, 1, ..., L-1\}$. $W_3^{(l)} \in \mathbb{R}^{d^{(l)} \times 4d^{(l)}}$, $W_4^{(l)} \in \mathbb{R}^{4d^{(l)} \times d^{(l+1)}}$, $b_3^{(l)} \in \mathbb{R}^{4d^{(l)}}$, and $b_4^{(l+1)} \in \mathbb{R}^{d^{(l+1)}}$ are learnable parameters.

4.4.2 Residual diffusion. We need to smooth the generated residual, as explained in Equation 2. Essentially, this smoothing kernel aggregates the residual information between the nodes. We employ the adaptive learning method to learn this diffusion kernel. As shown in the upper right part of Figure 3, we first randomly initialize a learnable kernel embedding $E_G \in \mathbb{R}^{N \times d_g}$. Then, we calculate the diffusion kernel with the adaptive learning strategy:

$$\mathcal{A}_d\left(\hat{\mathcal{K}}\right) = \text{Softmax}\left(\text{ReLU}\left(\hat{\mathcal{K}} - \text{diag}\left(\hat{\mathcal{K}}\right)\right)\right) \in [0,1]^{N \times N}, \tag{17}$$

$$\hat{\mathcal{K}} = E_G \times E_G^{\top} \in \mathbb{R}^{N \times N}, \tag{18}$$

where diag $(\cdot)$ is diagonal operator. The final diffusion kernel $\mathcal{K}$ can be computed:

$$\mathcal{K} = \boldsymbol{\beta}\left(\mathrm{I}_N + \boldsymbol{\alpha}\mathcal{A}_d\left(\hat{\mathcal{K}}\right)\right) \in \mathbb{R}^{N\times N}, \tag{19}$$

$$\boldsymbol{\alpha} = \mathrm{diag}\left(\alpha_1, \alpha_2, ..., \alpha_N\right) \in (-1,1)^{N\times N}, \tag{20}$$

$$\boldsymbol{\beta} = \mathrm{diag}\left(\beta_1, \beta_2, ..., \beta_N\right) \in (0,1)^{N\times N}, \tag{21}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are learnable parameters. Finally, we apply this kernel to smooth the residual representation with J finite steps:

$$\tilde{\mathrm{Z}}_{\mathrm{R}} = \mathcal{K}^J \times_2 \mathrm{Z}_{\mathrm{R}} \in \mathbb{R}^{T\times N\times d_c}. \tag{22}$$

4.4.3 Correction prediction. We use the generated residual term as input to the decoder to produce the corrected prediction:

$$\mathrm{Y}_{\mathrm{cor}} = \tilde{\mathrm{Z}}_{\mathrm{R}} W_c + b_c \in \mathbb{R}^{T_P\times N\times c}, \tag{23}$$

where $W_c \in \mathbb{R}^{(T\times d_c)\times(T_P\times c)}$ and $b_{\mathrm{out}} \in \mathbb{R}^{T_P\times c}$ are learnable parameters. Finally, we use the generated correction term Y_{cor} to correct the base prediction to produce the final prediction:

$$\hat{\mathrm{Y}} = \mathrm{Y}_{\mathrm{base}} + \mathrm{Y}_{\mathrm{cor}} \in \mathbb{R}^{T_P\times N\times c}. \tag{24}$$

Algorithm 1: BiST for spatiotemporal prediction

```
Input: Observed input $\mathrm{X} \in \mathbb{R}^{T\times N\times c}$; // No label required.
Output: Future prediction $\hat{\mathrm{Y}} \in \mathbb{R}^{T_P\times N\times c}$
# Preprocessing;
$\mathrm{X}_0 \leftarrow \mathrm{X}$ in Eq. 5 ~ 7;                 // Temporal decomp.
$\mathrm{Z}_{\mathrm{In}} \leftarrow \mathrm{X}_0, \boldsymbol{e}_P, \boldsymbol{e}_T, \boldsymbol{e}_D, \boldsymbol{e}_S$ in Eq. 8; // Input representation
# Forward spatiotemporal learning;
$\mathrm{Z}_{\mathrm{La}} \leftarrow \mathrm{Z}_{\mathrm{In}}$ in Eq. 9; // Label representation learning
$\mathrm{Y}_{\mathrm{base}} \leftarrow \mathrm{Z}_{\mathrm{La}}$ in Eq. 10;              // Base prediction
# Backward residual correction;
$\mathrm{Z}_{\mathrm{R}} \leftarrow E_q, E_k, \mathrm{Z}_{\mathrm{La}}$ in Eq. 11 ~ 15;   // Residual learning
$\tilde{\mathrm{Z}}_{\mathrm{corr}} \leftarrow \mathrm{Z}_{\mathrm{R}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, E_G$ in Eq. 16 ~ 22;          // Diffusion
$\mathrm{Y}_{\mathrm{corr}} \leftarrow \tilde{\mathrm{Z}}_{\mathrm{corr}}$ in Eq. 23;        // Correction prediction
# Final prediction;
$\hat{\mathrm{Y}} \leftarrow \mathrm{Y}_{\mathrm{base}} + \mathrm{Y}_{\mathrm{cor}}$ in Eq. 24;            // Final prediction
```

5 EXPERIMENT

In this section, we conduct a comprehensive evaluation of the proposed BiST. We will answer the following potential questions. **Q.1 and Q.2**. How does the model perform for short-term and long-term prediction tasks? **Q.3**. What is the computational complexity and memory usage of this model? **Q.4**. Is each component of the model valid? **Q.5**. How do hyperparameters affect model performance? **Q.6**. Can the model handle spatiotemporal deviation? **Q.7**. Can modeling residual dependencies across multiple time steps bring performance gains? **Q.8**. What interesting cases does BiST find?

5.1 Experiment Setting

5.1.1 Datasets. To evaluate the effectiveness of our model, we conduct a comprehensive experiment across 13 spatiotemporal datasets that covered the domains of traffic and atmospheric conditions. The statistical details of these datasets are provided in Table 2.

Among these, we include several large-scale datasets, with two featuring **large-scale datasets**—the XTraffic and CA—and one with a **very long-range dataset**—the XXLTraffic dataset. The XTraffic dataset [12] contains 16,972 nodes, and the CA dataset within the LargeST dataset [33] includes 8,600 nodes. To our knowledge, these are the two largest open-source datasets in the spatiotemporal domain regarding the number of nodes. We also select a dataset with an exceptionally large temporal scale, XXLTraffic [66], which records over 20 years of traffic data. For our experiments, we use the accessible sub-dataset, FULL-PeMS05. Additionally, the KnowAir [29] and LargeST datasets cover four and five years of data, respectively.

Table 2: Statistics of the used large spatiotemporal datasets. XXLTraffic does not provide a spatial adjacency matrix resulting in missing # edges. Data Points are the multiplication of nodes and samples. M: million (10^6). B: billion (10^9).

Dataset	# Nodes	# Edges	Time period	Data points
PeMS03 [50]	358	546	09/01/2018 ~ 11/30/2018	9.38M
PeMS04 [50]	307	338	01/01/2018 ~ 02/28/2018	5.22M
PeMS07 [50]	883	865	05/01/2017 ~ 08/06/2017	24.92M
PeMS08 [50]	170	276	07/01/2016 ~ 08/31/2016	3.04M
METR-LA [29]	207	1,515	03/01/2012 ~ 06/27/2012	7.09M
PeMS-Bay [29]	325	2,369	01/01/2017 ~ 06/30/2017	16.94M
KnowAir [60]	184	3,796	01/01/2015 ~ 12/31/2018	2.15M
SD [33]	716	17,319	01/01/2017 ~ 12/31/2021	0.38B
GBA [33]	2,352	61,246	01/01/2017 ~ 12/31/2021	1.24B
GLA [33]	3,834	98,703	01/01/2017 ~ 12/31/2021	2.02B
CA [33]	8,600	201,363	01/01/2017 ~ 12/31/2021	4.52B
XTraffic [12]	16,972	870,100	01/01/2023 ~ 12/31/2023	1.78B
XXLTraffic [66]	573	-	03/07/2005 ~ 03/20/2024	1.14B

5.1.2 Setting. We adopt the default code frame of LargeST in all datasets for a fair comparison. All data sets are divided into the training set, the validation set, and the test set in a ratio of 6:2:2 along the time axis. We adopt Adam [24] optimizer with a learning rate 0.002 and predefined milestones decay factor of 0.5. To evaluate the efficacy of our framework, we employ four common metrics, including Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The models are executed on a Nvidia A100 with 40GB memory, and the code environment is based on the PyTorch framework using Python 3.8.3. The XXLTraffic dataset is used to evaluate the long-term prediction performance of models. The other datasets are used for the short-term prediction task. For each experiment, we performed it five times and reported the average value for a comprehensive comparison.

Concretely, we use 3 layers of MLP in both forward and backward modules, i.e., $L = 3$. The finite steps in residual diffusion J are set to 4. The dimensions of all embeddings are equal to 32. The temporal decomposition kernel size k is equal to 3 in the short-term

Table 3: Short-term performance comparisons on on both the LargeST dataset, spanning a five-year period, and the XTraffic dataset. "*" means that we reduce the hyperparameter. The length of the input time window and future prediction window are both set to 12. The unit of MAPE is percent (%). We bold the best-performing model results in red and underline the sub-optimal model results in blue for each dataset.

	Method	Horizon 3			Horizon 6			Horizon 12			Average		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
SD	LSTM	18.64±0.34	29.27±0.58	11.52±0.81	24.88±0.57	39.15±0.64	16.62±1.24	35.93±0.98	55.44±0.73	25.17±1.39	25.27±0.71	39.44±0.62	17.10±1.28
	DCRNN	18.48±0.38	29.04±0.34	11.22±0.27	24.48±0.52	38.77±0.64	16.16±0.29	35.52±0.83	55.18±0.92	25.07±0.42	25.02±0.57	38.95±0.71	16.65±0.33
	STGCN	18.95±0.92	29.04±0.22	12.84±0.48	21.76±1.46	35.08±0.12	14.50±0.22	26.74±0.21	43.04±0.45	18.06±0.54	21.80±0.06	34.37±0.36	14.79±1.30
	GWNet	16.84±0.48	26.64±1.08	11.05±0.59	20.67±1.03	33.46±1.19	13.67±0.64	26.32±1.38	42.65±1.33	17.45±0.80	20.85±1.14	32.97±1.12	13.99±0.79
	STNorm	16.44±0.29	25.91±0.44	11.12±0.28	20.30±0.55	32.88±0.63	13.86±0.54	25.53±0.62	41.09±0.99	17.63±1.01	20.88±0.84	32.73±0.67	13.71±0.69
	STID	17.22±0.39	28.10±0.37	10.79±1.02	21.07±0.47	34.89±0.56	14.23±1.41	27.02±0.76	44.18±1.03	18.25±1.46	20.79±0.55	34.34±0.73	14.21±1.36
	LarST	17.14±0.25	27.55±0.19	11.47±0.27	22.57±0.72	34.30±0.20	14.58±1.37	27.65±1.12	43.36±0.19	18.32±1.36	22.27±0.13	34.55±1.06	15.19±0.45
	STGODE	16.77±0.72	27.93±0.54	11.75±0.86	22.88±1.21	35.94±0.69	14.94±1.26	26.73±1.32	45.91±0.84	18.69±1.38	21.13±1.08	34.03±0.95	14.98±1.17
	ASTGCN	18.61±1.23	29.11±1.24	14.09±1.26	25.05±1.96	39.09±1.33	17.80±1.96	33.63±2.31	50.42±3.09	25.27±2.53	25.55±2.16	39.12±2.07	18.12±1.73
	AGCRN	16.63±0.25	26.98±1.13	11.09±0.43	20.43±0.76	32.95±1.31	13.85±0.51	25.27±1.39	40.26±1.53	17.09±0.63	20.66±0.79	32.81±1.28	13.81±0.58
	DSTAGNN	18.47±1.26	28.93±1.28	11.14±1.23	24.77±1.55	38.82±1.49	16.45±1.39	35.52±1.66	55.23±2.19	24.94±1.54	24.79±1.52	39.24±1.72	16.89±1.46
	STAEformer	17.11±0.29	26.76±0.55	12.42±0.61	20.78±0.98	33.31±0.81	13.91±0.93	26.12±1.37	41.55±1.50	18.15±1.34	21.02±0.95	33.41±0.99	14.71±1.12
	STTN	16.91±0.31	27.66±0.35	11.35±0.22	22.50±0.33	35.70±0.39	14.54±0.34	26.58±0.87	45.78±0.88	18.45±0.60	20.73±0.52	33.67±0.51	14.78±0.43
	DGCRN	16.17±0.26	26.72±0.46	11.13±0.23	20.00±0.49	31.99±0.44	12.74±0.23	25.08±0.54	40.08±0.65	17.49±0.57	19.93±0.48	32.02±0.55	13.19±0.48
	DDGCRN	16.28±0.48	26.63±0.65	10.31±0.39	19.92±0.54	32.01±0.84	12.98±0.41	25.07±0.75	39.94±1.06	17.63±0.47	19.99±0.68	32.08±0.82	13.17±0.42
	D^{2}STGNN	15.99±0.43	26.55±0.33	10.17±0.23	19.87±0.79	31.77±0.42	12.72±0.31	24.98±0.98	39.91±0.55	17.37±0.76	19.92±0.85	31.99±0.51	13.09±0.45
	Ours	**15.06±0.32**	**24.96±0.28**	**10.12±0.15**	**18.39±0.34**	**30.73±0.39**	**12.39±0.26**	**23.81±0.52**	**38.73±0.62**	**15.92±0.38**	**18.30±0.37**	**30.44±0.42**	**12.37±0.31**
GBA	LSTM	17.37±0.34	28.25±0.74	10.78±0.22	23.45±1.04	38.12±1.13	15.52±0.56	34.32±1.41	52.92±1.75	22.64±0.62	23.86±1.23	38.01±1.46	15.64±0.44
	DCRNN	17.21±0.25	27.86±0.65	10.39±0.49	23.09±0.31	37.85±0.87	15.10±0.55	33.90±0.76	52.77±1.02	22.33±0.67	23.68±0.45	37.56±0.91	15.33±0.53
	STGCN	19.45±0.72	30.38±1.06	13.92±1.22	24.64±0.75	38.86±1.23	17.14±1.37	30.98±1.11	46.87±1.47	20.49±1.89	24.84±0.84	37.68±1.27	16.59±1.48
	GWNet	17.79±0.36	28.05±0.97	10.54±0.27	22.91±0.81	35.72±1.13	13.58±0.41	29.32±1.23	44.81±1.27	18.32±0.66	23.03±1.04	35.36±1.19	13.91±0.48
	STNorm	17.44±0.39	27.79±0.31	11.23±0.33	23.05±0.62	36.42±0.41	14.71±0.64	30.85±0.72	46.15±1.43	20.98±0.92	23.25±0.23	35.49±0.94	15.56±0.68
	STID	17.34±0.15	28.65±0.27	11.39±0.21	22.82±0.96	36.34±0.71	14.76±0.89	29.82±1.08	46.14±1.32	20.12±1.30	22.97±0.75	36.24±0.92	14.47±0.82
	LarST	18.39±0.39	30.26±0.42	11.63±0.26	23.85±0.49	39.32±1.03	15.17±0.72	31.47±1.54	46.83±2.85	21.22±0.91	23.91±1.05	36.92±1.47	15.36±0.83
	STGODE	17.39±0.54	28.71±0.26	11.50±0.19	22.69±1.11	36.24±1.42	14.73±0.48	30.55±1.49	47.67±1.88	20.63±1.04	23.38±1.26	36.16±1.64	14.78±0.76
	ASTGCN	17.77±1.26	29.55±1.25	11.48±0.71	25.54±2.15	40.05±2.36	16.54±1.05	37.72±2.98	57.68±2.91	24.92±1.48	26.28±2.37	41.14±2.38	17.35±1.24
	AGCRN	18.02±0.64	28.78±0.99	11.66±0.95	22.73±1.16	37.02±1.65	14.36±1.19	29.65±1.48	45.49±1.85	20.03±1.38	23.12±1.11	36.24±1.67	15.09±1.20
	DSTAGNN	17.69±1.42	29.47±1.56	11.40±1.45	25.51±1.45	40.06±1.65	16.38±1.78	37.57±1.59	57.64±2.43	24.83±2.65	26.36±1.44	41.15±1.83	17.44±1.62
	STAEformer	18.41±0.45	29.02±0.34	12.22±0.14	23.68±0.52	36.81±1.35	15.07±0.24	31.03±0.75	46.09±0.17	20.89±0.27	23.57±0.68	36.06±1.48	15.58±0.19
	STTN	17.35±0.17	28.61±0.35	11.11±0.22	22.22±0.29	35.92±0.44	14.36±0.25	30.28±0.64	47.46±0.84	20.22±0.52	23.14±0.58	35.92±0.57	14.62±0.43
	DGCRN	17.26±0.28	29.18±0.36	10.57±0.25	23.08±0.57	37.89±0.39	14.74±0.53	29.84±0.73	46.72±0.57	18.65±0.62	23.05±0.62	36.58±0.44	13.93±0.47
	DDGCRN	17.48±0.23	29.34±0.30	10.82±0.54	23.12±0.56	37.87±0.47	14.75±0.59	29.76±0.64	46.93±0.52	18.52±0.68	22.94±0.67	36.36±0.41	13.94±0.61
	D^{2}STGNN	17.23±0.46	29.11±0.59	10.52±0.27	22.75±0.73	37.73±0.88	14.48±0.33	29.55±1.13	46.69±1.08	18.38±0.43	22.65±0.86	36.36±0.73	13.92±0.37
	Ours	**15.42±0.12**	**26.02±0.38**	**9.88±0.17**	**20.41±0.37**	**33.59±0.69**	**12.86±0.29**	**26.77±0.85**	**42.59±1.05**	**17.68±0.58**	**20.30±0.53**	**33.26±0.83**	**13.12±0.34**
GLA	LSTM	18.04±0.19	29.62±0.26	11.43±0.33	25.19±1.38	41.32±2.73	17.07±1.29	36.17±2.48	56.29±3.12	25.72±2.15	25.36±1.44	41.18±2.49	17.22±1.74
	DCRNN	17.78±0.57	29.37±0.36	11.33±0.60	24.60±1.45	41.18±1.35	16.71±1.51	35.95±2.24	56.00±2.38	25.39±2.67	24.87±1.60	41.08±1.35	16.89±1.65
	STGCN	20.14±1.12	30.73±0.53	13.93±0.88	24.93±1.27	37.96±1.07	17.29±1.38	31.62±1.89	49.22±2.19	22.07±1.88	24.86±1.66	38.37±1.06	16.93±1.08
	GWNet	17.45±0.46	28.39±0.78	11.98±0.35	23.25±0.54	35.85±1.34	15.92±0.99	30.92±1.38	47.40±1.69	21.35±1.16	22.73±0.84	35.72±1.47	15.89±0.63
	STNorm	18.14±0.23	28.52±0.52	11.64±0.24	22.77±0.37	36.65±1.09	14.61±0.65	29.92±0.75	45.74±1.68	19.97±1.49	22.65±0.63	36.18±1.14	15.23±0.89
	STID	18.01±0.11	28.94±0.47	12.15±0.17	23.56±0.78	37.53±0.76	16.22±0.63	31.61±1.41	49.27±1.01	21.99±0.66	23.95±0.65	37.85±0.75	16.34±0.44
	LarST	19.31±0.52	29.67±0.89	12.73±0.71	23.78±0.88	37.68±1.08	16.95±1.11	32.43±1.34	50.54±1.41	21.76±1.36	24.34±1.02	37.26±1.20	16.67±1.16
	STGODE	18.39±0.53	28.96±1.04	12.88±0.69	23.85±1.13	37.57±1.62	17.29±1.04	32.09±1.25	50.61±2.76	21.91±2.18	23.84±0.91	37.51±1.43	16.81±1.61
	ASTGCN	20.14±0.94	32.28±1.06	15.89±1.47	28.23±2.04	44.56±2.48	19.94±1.42	40.78±2.94	61.65±3.57	32.71±2.71	28.53±2.37	44.42±2.43	21.81±3.24
	AGCRN	17.24±0.35	28.01±0.30	11.38±0.15	22.21±0.79	35.71±0.74	14.15±0.35	29.21±1.23	45.67±0.81	19.59±0.57	22.34±0.88	35.38±0.69	14.43±0.37
	DSTAGNN	20.04±1.28	32.12±1.25	15.73±1.26	28.22±2.14	44.41±1.32	19.90±1.56	35.79±2.23	51.58±2.57	32.84±1.82	28.41±1.63	44.24±2.29	21.77±1.44
	STAEformer*	18.87±0.71	29.92±0.69	11.38±0.23	24.25±0.92	37.15±0.77	14.99±0.51	30.55±1.37	45.70±0.95	20.42±1.14	23.73±1.03	36.58±0.85	15.35±0.78
	STTN*	19.05±0.43	29.88±0.28	12.45±0.38	24.65±0.60	37.46±0.32	16.94±0.61	31.68±0.87	50.51±0.55	21.50±0.92	23.42±0.73	37.07±0.42	16.33±0.65
	DGCRN*	19.15±0.39	30.65±0.51	12.11±0.23	25.60±0.47	39.94±1.28	15.92±0.67	34.31±1.56	50.16±1.49	22.36±1.29	25.80±0.62	39.58±1.07	15.73±0.52
	DDGCRN*	19.52±0.59	30.83±0.68	12.31±0.24	25.55±0.71	39.94±1.25	15.83±0.64	34.41±1.38	50.13±2.17	22.08±1.57	25.91±0.87	39.50±1.26	15.95±0.75
	D^{2}STGNN*	19.56±0.93	30.86±0.91	12.39±0.28	25.83±1.01	40.09±1.01	16.09±1.04	34.51±1.69	50.35±1.88	22.37±1.54	26.16±1.35	39.66±1.14	15.96±1.10
	Ours	**16.20±0.15**	**26.91±0.29**	**10.46±0.18**	**20.38±0.48**	**33.64±0.78**	**12.72±0.43**	**26.67±0.86**	**42.92±1.05**	**17.81±0.52**	**20.36±0.58**	**33.48±0.62**	**13.22±0.39**
CA	LSTM	17.07±0.98	27.96±1.18	11.96±0.44	23.43±1.34	38.23±1.48	16.22±0.72	33.83±2.29	53.52±2.13	25.55±0.92	23.74±1.73	38.15±1.45	17.38±0.87
	DCRNN	16.95±0.64	27.59±1.42	11.69±0.39	23.18±1.50	37.86±1.47	15.73±0.56	33.66±1.54	53.15±2.23	25.08±1.34	23.38±1.28	37.66±1.30	16.91±0.62
	STGCN	18.49±1.08	30.24±1.32	13.69±0.26	22.71±1.15	36.84±1.35	16.97±0.35	28.72±1.31	46.25±1.42	21.25±0.48	22.58±1.19	36.29±1.38	16.88±0.32
	GWNet	16.22±0.43	26.53±0.37	11.76±0.28	20.69±0.75	33.67±0.46	14.32±0.37	27.48±1.14	42.84±0.59	20.79±0.41	20.52±0.82	33.94±0.48	15.34±0.37
	STNorm	15.98±0.29	26.65±0.48	12.13±0.40	21.06±0.38	34.06±0.71	15.23±0.53	27.25±0.54	42.82±0.86	20.29±0.61	21.04±0.39	32.96±0.54	15.24±0.49
	STID	16.21±0.32	26.97±0.72	11.75±0.48	21.49±0.36	34.61±1.19	15.18±0.66	28.05±0.59	44.89±1.25	20.64±0.81	21.62±0.46	34.63±0.91	15.64±0.68
	LarST	16.26±0.21	27.10±0.95	11.68±0.29	21.49±0.42	34.24±1.03	14.95±0.43	27.78±0.83	44.36±1.08	20.71±0.75	21.09±0.54	34.01±1.02	15.42±0.56
	STGODE	18.33±0.34	29.39±1.01	12.89±0.82	24.21±1.29	37.63±1.24	17.31±1.27	32.38±1.36	50.88±1.48	21.61±1.76	24.42±1.07	37.58±1.41	16.99±1.25
	ASTGCN*	18.77±1.29	29.63±1.03	15.85±1.14	25.97±2.15	42.29±1.32	17.16±1.81	38.93±2.49	56.92±1.82	28.67±2.17	27.33±1.69	42.59±1.63	19.77±1.75
	AGCRN*	16.82±0.44	28.43±0.44	12.47±0.28	21.42±0.64	34.96±0.80	15.77±0.68	28.12±0.72	44.07±1.31	21.54±0.77	21.36±0.58	34.79±1.15	15.94±0.44
	DSTAGNN*	17.70±0.81	28.81±0.37	13.17±0.67	22.51±1.47	36.77±0.52	16.83±1.46	29.55±1.65	45.59±1.17	22.92±1.49	22.64±0.99	36.18±0.87	16.87±1.15
	STAEformer*	17.87±0.24	29.01±0.83	13.29±0.35	22.89±0.98	37.15±1.32	17.08±0.79	29.97±1.78	45.96±1.41	23.36±1.21	22.89±1.25	36.51±1.24	17.33±0.98
	Ours	**15.38±0.32**	**25.53±0.22**	**10.88±0.17**	**19.93±0.36**	**32.75±0.42**	**13.92±0.31**	**26.86±0.59**	**41.97±0.73**	**19.63±0.42**	**19.95±0.42**	**32.67±0.53**	**14.21±0.24**
XTraffic	LSTM	11.10±0.64	21.47±0.28	16.25±0.26	15.31±0.48	28.82±1.31	22.77±1.52	21.97±1.31	40.75±1.54	30.11±2.37	15.56±0.59	29.14±1.24	22.13±1.42
	DCRNN	10.99±0.25	21.14±0.32	15.93±0.37	14.83±0.39	28.62±1.64	22.66±1.37	21.53±1.42	40.62±2.29	29.79±1.59	15.10±0.63	28.97±1.21	22.02±1.03
	STGCN	13.67±0.19	25.20±0.65	19.02±0.45	15.28±0.55	28.11±0.72	19.94±0.56	17.98±0.93	33.54±0.84	22.59±0.92	15.27±0.66	28.49±0.72	20.41±0.76
	GWNet*	11.79±0.20	21.89±0.29	16.24±0.25	15.93±0.36	29.57±0.89	22.82±0.36	22.37±0.45	40.59±1.74	32.42±1.39	16.06±0.37	29.85±1.13	23.52±0.92
	STNorm	9.84±0.39	18.71±0.53	16.30±0.29	11.62±0.41	22.19±0.71	18.12±0.43	14.26±0.54	27.08±1.03	22.54±0.86	11.57±0.43	22.11±0.87	18.44±0.89
	STID	10.05±0.33	19.04±0.78	15.14±0.16	12.02±0.49	22.77±1.15	17.32±0.21	14.42±0.86	28.25±1.54	21.36±0.39	11.71±0.74	22.81±0.98	17.54±0.24
	LarST	10.94±0.24	19.86±0.75	17.14±0.42	12.35±1.13	23.48±1.01	18.84±1.19	16.68±1.25	31.85±1.65	23.36±1.27	12.69±0.82	24.54±1.06	18.77±0.74
	STGODE*	11.13±0.88	20.69±0.37	19.92±0.97	13.15±1.05	24.57±1.04	23.67±1.33	16.93±1.46	32.02±1.06	30.46±1.64	13.37±1.15	24.92±1.32	24.44±1.41
	Ours	**8.87±0.11**	**17.62±0.22**	**13.83±0.26**	**10.48±0.24**	**20.79±0.38**	**16.12±0.34**	**12.77±0.46**	**25.45±0.55**	**19.88±0.47**	**10.45±0.35**	**20.74±0.47**	**16.21±0.38**

prediction, 5 in the KnowAir dataset, and 25 in long-term prediction. The number of virtual nodes K is set to 8 in SD, 24 in GBA, 32 in GLA, 64 in CA, 128 in XTraffic, and 3 in KnowAir and XXLTraffic.

5.1.3 Baselines. We compare two types of model: spatiotemporal models and time series models excelling at long-term predictions. **Spatiotemporal models** include DCRNN [29], STGCN [67], GWNet [64], STNorm [5], STID [47], LarST [56], STGODE [8], ASTGCN [16], AGCRN [1], DSTAGNN [25], STAEformer [32], STTN [65], DGCRN [27], DDGCRN [62] and D^{2}STGNN [49]. **Time series models** contain DLinear [71], Mamba [13], Autoformer [63], iTransformer [34], DSformer [68], TimeMixer [61], SparseTSF [31], UMixer [39], CATS [36], SOFTS [17] and CrossGNN [20].

5.2 Short-term Prediction Performance Comparison (Q.1)

We set both the input and prediction windows to 12 to evaluate the short-term prediction performance of each model.

As shown in Table 3 and Table 4, BiST consistently demonstrates superior performance across all forecasting horizons on these large-scale spatiotemporal datasets, highlighting the effectiveness of our model in handling numerous spatiotemporal data. Conversely, HL exhibits the poorest performance, probably due to the volatility of temporal data. Despite LSTM being a classical recurrent neural network for sequence data and its lack of spatial influence learning, which is critical in spatiotemporal modeling, and surprisingly remains highly competitive in short-term predictions when substantial amounts of data are available. STGCN and GWNet, the pioneering works that integrate GNN with gated TCN, achieve promising performance even compared to many recent works, such as ASTGCN. STGODE improves model accuracy by solving continuous layers of GNN as a replacement. AGCRN replaces the fully connected layer in GRU [3] with an adaptive diffusion matrix from GWNet. STID employs learnable node embeddings to characterize the spatiotemporal structure, assisting MLP in learning, and showing good result stability on datasets with large spatial scale. STAEformer modifies the MLP structure in STID to a vanilla Transformer [51] architecture for temporal and spatial dimensions, but its quadratic complexity concerning the number of nodes limits its scalability to larger datasets. D^{2}STGNN models temporal and spatial dependencies with dynamic spatial topology and a decoupled spatiotemporal framework, performing better on smaller datasets, while STNorm enhances spatiotemporal learning through specialized normalization techniques, performing well on larger datasets.

Nevertheless, our model achieves dominant short-term forecast performance. In Table 3, BiST achieves a relative improvement of over 5% in most metrics. In about 20% of the metrics, BiST exhibites similar or even more than 10% relative improvement, with the maximum relative improvement reaching 12.28%.

5.3 Long-term Prediction Performance Comparison (Q.2)

We evaluate the long-term prediction performance of the models on the XXLTraffic dataset, which spans a very long period. To assess its ability to handle different temporal granularities, we aggregated the data into hourly and daily time scales. We compare our model against advanced spatiotemporal graph prediction models (such as STID and STAEformer) as well as long-term time series models. The official paper reports metrics on standardized data, and for intuitive comparison, we maintain this setup.

As shown in Table 5, for data with smaller temporal granularity, the STID model gains advantages by accurately modeling complex spatiotemporal correlations. However, for daily frequency data, long-series time models, such as SOFTS and TimeMixer, demonstrate superior prediction performance. This is primarily because daily data often exhibit strong periodicity, making the accurate modeling of these patterns essential, an area where these time series prediction models excel. For instance, Autoformer employs decomposition techniques alongside an autocorrelation mechanism, effectively capturing periodic patterns. Its performance surpasses that of the latest iTransformer, which utilizes attention and feedforward networks applied to the inverted dimension. Similarly, SOFTS adopts a centralized strategy to model dependencies among different variable channels, thereby achieving enhanced performance.

Our model demonstrates leading performance across various time scales in long-term forecasting, attributed to its effective utilization of spatial information and the implementation of its temporal decoupling module. We achieve a maximum relative improvement of 12.74%, with most metrics reflecting gains of more than 5%.

5.4 Model Efficiency Analysis (Q.3)

We compare the complexity of our proposed model with several advanced spatiotemporal prediction models. Using the GBA, CA, and XTraffic datasets as examples, we report the total training time, per-epoch training time, inference time, and memory usage, as shown in Table 6 and Figure 4. We can observe that STGCN and GWNet exhibit higher efficiency due to their utilization of TCN as a temporal module, enabling efficiency improvements through parallel strategies. While models from the Transformer family, D^{2}STGNN, and STAEformer demonstrate good predictive performance, the Transformer models consume significant computational time, leading to lower operational efficiency.

Regarding memory utilization, these models tend to stack neural network layers to enhance representational capacity. During the forward learning process, devices need to maintain an embedding vector for each node. When backpropagating errors, the regression loss function necessitates maintaining the computation cache for the entire graph, resulting in a substantial memory burden.

In contrast, our proposed model is based on lightweight MLP architecture, reducing time consumption. Furthermore, our model comprises only forward and backward modules, thereby reducing memory usage.

5.5 Ablation Study (Q.4)

We conduct an ablation study to explore the effectiveness of each component in BiST. "w/o tems" removes the temporal decoupling Technology, and "w/o tememb" and "w/o Noe" mean that we remove the temporal and node embeddings respectively. "w/o prompt" eliminates the spatiotemporal embedding prompt, "w/o back" uses only the base predictions from the forward process as the final prediction, omitting subsequent decoupling and residual correction modules, "w/o dec" means that we use two-layer MLP layers to

Table 4: Short-term performance comparisons on six traditional spatiotemporal datasets. The length of the input time window and the future prediction window is set to 12 for all datasets except KnowAir, where the length of both windows is 24. The performance reported is computed by averaging over all predicted time steps. The unit of MAPE is percent (%).

Dataset	PeMS03			PeMS04			PeMS07			PeMS08			METR-LA			KnowAir		
Method	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
LSTM	21.18±0.28	35.07±0.16	23.32±0.18	27.14±0.15	41.81±0.34	18.34±0.06	30.08±0.38	45.94±1.31	13.24±0.24	22.21±0.29	34.01±0.25	14.32±0.12	3.55±0.11	7.13±0.04	10.19±0.13	24.39±0.73	35.24±0.44	55.35±3.02
DCRNN	18.15±0.14	30.32±0.51	18.81±0.18	21.24±0.13	33.46±0.13	14.26±0.24	25.24±0.08	38.64±0.56	11.73±0.11	16.89±0.32	26.39±0.06	10.98±0.04	3.15±0.02	6.24±0.01	8.59±0.05	22.03±0.82	32.66±1.01	55.16±2.26
STGCN	17.39±0.11	28.87±0.33	17.11±0.12	20.03±0.11	31.76±0.59	13.23±0.15	21.64±0.07	34.85±0.46	13.98±0.22	15.64±0.36	25.16±0.07	10.37±0.09	3.11±0.02	6.25±0.04	8.62±0.08	22.49±0.85	31.83±0.59	52.16±2.32
GWNet	16.85±0.18	27.58±0.17	16.11±0.07	19.03±0.12	30.45±0.56	13.19±0.16	21.51±0.15	34.35±0.24	10.11±0.03	18.02±0.53	27.86±0.02	9.36±0.02	3.03±0.02	6.04±0.03	8.21±0.04	22.45±0.84	31.59±0.72	53.17±2.48
STNorm	15.42±0.17	25.82±0.26	14.67±0.04	19.48±0.14	32.36±0.36	12.24±0.25	20.49±0.02	34.82±0.52	8.58±0.05	15.57±0.52	24.95±0.06	10.05±0.03	3.14±0.01	6.41±0.02	8.72±0.07	23.02±0.81	32.85±0.59	52.77±2.85
STID	15.18±0.12	25.96±0.11	16.24±0.04	18.59±0.12	30.25±0.11	12.42±0.13	19.54±0.02	32.86±0.24	8.29±0.03	14.23±0.44	23.44±0.08	9.29±0.08	3.20±0.03	6.57±0.01	9.16±0.01	21.96±0.75	30.51±0.43	49.95±1.41
STGODE	16.39±0.14	27.94±0.15	16.79±0.09	20.99±0.05	32.79±0.35	13.57±0.14	22.99±0.04	37.59±0.63	10.23±0.11	16.71±0.42	25.88±0.02	10.59±0.08	3.12±0.01	6.27±0.04	8.97±0.13	21.46±0.78	31.51±0.59	48.47±1.95
ASTGCN	17.92±0.26	29.46±0.24	19.18±0.06	22.99±0.12	35.03±0.34	16.59±0.23	28.06±0.15	42.66±1.21	13.82±0.29	18.64±0.34	28.18±0.14	12.88±0.12	5.04±0.02	10.61±0.13	9.53±0.02	22.96±0.58	32.62±0.81	53.54±2.92
AGCRN	16.03±0.09	28.56±0.42	15.75±0.14	19.69±0.05	32.25±0.14	12.92±0.13	20.83±0.03	34.72±0.73	8.91±0.06	15.67±0.29	25.08±0.08	10.26±0.08	3.14±0.03	6.38±0.03	8.82±0.13	23.88±0.86	32.94±0.44	59.03±1.40
DSTAGNN	15.81±0.05	27.27±0.12	15.62±0.04	19.24±0.06	31.41±0.19	12.89±0.17	21.43±0.15	34.54±0.21	9.04±0.04	15.58±0.39	24.77±0.02	9.87±0.01	3.17±0.04	6.37±0.03	8.61±0.13	22.93±0.92	32.91±0.44	55.81±2.61
STAEformer	15.51±0.08	27.45±0.22	15.23±0.18	18.13±0.09	30.01±0.11	11.94±0.16	19.62±0.15	33.44±0.84	8.29±0.04	13.98±0.26	23.98±0.07	9.13±0.02	3.02±0.04	6.07±0.03	8.34±0.05	22.79±0.58	32.27±0.55	48.91±2.18
STTN	15.85±0.15	28.13±0.28	15.26±0.09	18.83±0.13	30.94±0.39	12.31±0.07	20.13±0.13	34.21±0.22	8.45±0.07	14.79±0.52	25.08±0.08	9.37±0.01	3.13±0.03	6.17±0.03	8.59±0.04	23.95±0.64	33.52±0.45	60.12±1.74
DGCRN	15.71±0.12	27.46±0.27	15.12±0.15	19.68±0.13	31.47±0.44	13.57±0.24	20.84±0.19	34.13±0.46	9.51±0.08	15.11±0.48	24.11±0.06	9.96±0.06	3.11±0.03	6.22±0.04	8.67±0.05	22.47±0.62	32.49±0.67	54.78±1.89
DDGCRN	14.76±0.21	25.11±0.37	14.33±0.06	18.46±0.11	30.53±0.56	12.25±0.14	19.74±0.18	33.03±0.22	8.43±0.09	14.48±0.11	23.76±0.02	9.82±0.02	3.04±0.02	6.07±0.04	8.49±0.03	21.54±0.96	31.07±0.92	51.77±2.11
D^{2}STGNN	14.61±0.07	25.05±0.26	14.39±0.11	18.55±0.08	30.75±0.19	12.07±0.08	19.80±0.08	33.08±0.72	8.41±0.09	14.42±0.43	23.82±0.07	9.35±0.02	3.01±0.03	6.05±0.02	8.41±0.04	21.49±0.55	30.42±0.61	49.54±2.69
Ours	**14.33±0.05**	**24.29±0.16**	**14.19±0.07**	**17.95±0.09**	**29.56±0.12**	**11.93±0.12**	**19.23±0.03**	**32.59±0.15**	**8.08±0.05**	**13.78±0.16**	**23.32±0.05**	**8.94±0.03**	**2.97±0.01**	**6.02±0.02**	**8.14±0.03**	**20.27±0.51**	**29.75±0.54**	**47.29±1.41**

Table 5: Average long-term prediction performance on XXL-Traffic dataset. "Hourly" and "daily" are the sampling frequencies used in practice. The length of the input window is 96 with prediction window lengths of {96, 192, 336}.

	XXLTraffic	Horizon 96		Horizon 192		Horizon 336	
		MSE	MAE	MSE	MAE	MSE	MAE
Hourly	STID	0.046±0.002	0.124±0.004	0.052±0.002	0.131±0.002	0.055±0.005	0.141±0.004
	STAEformer	0.046±0.001	0.130±0.004	0.053±0.005	0.133±0.005	0.059±0.005	0.153±0.004
	Dlinear	0.054±0.005	0.187±0.014	0.062±0.001	0.169±0.002	0.061±0.003	0.171±0.004
	Mamba	0.045±0.002	0.161±0.003	0.056±0.005	0.154±0.002	0.054±0.002	0.152±0.006
	Autoformer	0.055±0.005	0.215±0.011	0.074±0.004	0.211±0.015	0.077±0.009	0.216±0.014
	iTransformer	0.083±0.008	0.255±0.012	0.102±0.005	0.244±0.013	0.101±0.014	0.253±0.013
	DSformer	0.067±0.007	0.158±0.004	0.073±0.004	0.159±0.001	0.071±0.004	0.156±0.003
	TimeMixer	0.064±0.007	0.156±0.004	0.074±0.003	0.170±0.010	0.074±0.005	0.171±0.001
	SparseTSF	0.114±0.009	0.192±0.007	0.099±0.009	0.173±0.001	0.099±0.012	0.174±0.008
	Umixer	0.082±0.008	0.181±0.009	0.074±0.002	0.162±0.007	0.072±0.009	0.170±0.006
	CATS	0.056±0.003	0.139±0.008	0.060±0.004	0.141±0.003	0.062±0.007	0.143±0.009
	SOFTS	0.068±0.001	0.165±0.003	0.078±0.002	0.175±0.008	0.087±0.002	0.187±0.005
	CrossGNN	0.111±0.007	0.206±0.016	0.097±0.006	0.191±0.002	0.098±0.008	0.197±0.006
	Ours	**0.041±0.002**	**0.114±0.003**	**0.046±0.004**	**0.121±0.003**	**0.051±0.005**	**0.127±0.004**
Daily	STID	0.178±0.004	0.259±0.003	0.217±0.003	0.306±0.002	0.251±0.003	0.332±0.004
	STAEformer	0.184±0.006	0.274±0.003	0.221±0.006	0.317±0.004	0.275±0.002	0.359±0.005
	Dlinear	0.166±0.003	0.238±0.005	0.209±0.003	0.282±0.002	0.242±0.003	0.298±0.004
	Mamba	0.177±0.012	0.254±0.011	0.238±0.013	0.314±0.014	0.293±0.013	0.327±0.013
	Autoformer	0.177±0.006	0.259±0.002	0.222±0.009	0.275±0.003	0.249±0.004	0.307±0.014
	iTransformer	0.176±0.004	0.255±0.003	0.232±0.002	0.303±0.011	0.256±0.003	0.309±0.012
	DSformer	0.176±0.009	0.250±0.004	0.224±0.003	0.282±0.007	0.252±0.003	0.296±0.003
	TimeMixer	0.158±0.003	0.232±0.004	0.204±0.004	0.275±0.001	0.236±0.001	0.296±0.005
	SparseTSF	0.165±0.009	0.239±0.006	0.210±0.008	0.279±0.006	0.246±0.002	0.294±0.007
	Umixer	0.165±0.004	0.240±0.002	0.211±0.005	0.283±0.009	0.240±0.002	0.296±0.004
	CATS	0.175±0.005	0.246±0.008	0.251±0.013	0.317±0.015	0.275±0.012	0.334±0.015
	SOFTS	0.156±0.002	0.214±0.001	0.204±0.011	0.259±0.005	0.242±0.008	0.296±0.002
	CrossGNN	0.163±0.005	0.235±0.008	0.207±0.001	0.276±0.009	0.243±0.008	0.295±0.008
	Ours	**0.147±0.004**	**0.207±0.010**	**0.178±0.005**	**0.245±0.009**	**0.224±0.003**	**0.288±0.004**

Table 6: Efficiency comparison of all models when achieving optimal performance on SD dataset.

SD Method	Performance (MAE)	Training (s/epoch)	Inference (s)	Total (hour)	Memory (MB)	Batch Size
STGCN [64]	21.80±0.06	133.3	31.8	2.8	3,452	64
GWNet [64]	20.85±1.14	321.9	45.5	10.5	7,978	64
STNorm [5]	20.88±0.84	97.9	20.1	2.43	3,762	64
STGODE [8]	21.13±1.08	498.0	81.1	8.1	18,948	64
ASTGCN [16]	25.55±2.16	493.8	84.9	11.9	8,984	64
AGCRN [1]	20.66±0.79	380.8	53.7	10.8	8,116	64
STAEformer [32]	21.02±0.95	242.7	26.6	4.6	40,822	55
D^{2}STGNN [49]	19.92±0.85	2,320.5	324.7	42.8	40,270	31
Ours	**18.30±0.37**	79.6	15.05	0.8	2,965	64

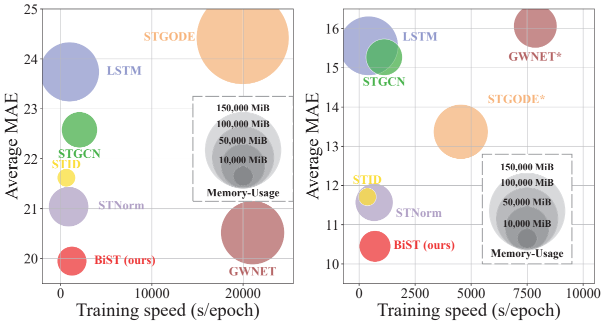


Figure 4: Efficiency comparison of optimal performance on CA and XTraffic datasets.

replace the spatiotemporal decoupling residual learning module, and "w/o adp" removes adaptive diffusion kernel learning and uses an identity matrix.

As shown in Table 7, results demonstrate that each component of the model is effective. The 'w/o prompt' variant exhibits lower prediction performance, indicating that integrating various prior knowledge enhances prediction accuracy. The 'w/o back' variant performs poorest, highlighting the importance of the backward correction process. The 'w/o dec' variant shows higher prediction errors, suggesting that decomposing spatiotemporal features into personalized and contextual features benefits inconsistent information modeling. The suboptimal prediction performance of "w/o prompt" validates that embedding prior knowledge can better guide model learning. In summary, ablation experiments on three datasets demonstrate that each of the involved components is effective.

5.6 Hyperparameter Sensitivity Analysis (Q.5)

In this section, we will analyze the impact of four key hyperparameters in the SD dataset. The results are shown in Figure 5.

Residual diffusion layers J. When J is equal to 4 in Equation 22, BiST achieves the best performance. A small number of residual propagation layers may not fully capture the residual information, while a large number of propagation steps can lead to oversmoothing commonly seen in GNNs, resulting in performance degradation.

Table 7: Ablation experiments on three datasets.

Method		**Ours**	w/o tems	w/o tememb	w/o Noe	w/o prompt	w/o back	w/o dec	w/o adp
SD	MAE	**18.30±0.37**	19.30±0.41	20.39±0.54	20.39±0.42	20.40±0.73	20.71±0.97	19.59±0.56	19.53±0.89
	RMSE	**30.44±0.42**	31.13±0.78	32.73±0.59	33.07±0.51	34.42±1.19	34.11±1.65	32.16±0.76	31.22±0.71
	MAPE	**12.37±0.31**	12.42±0.29	13.79±0.42	13.45±0.43	13.80±0.42	13.42±0.46	12.84±0.47	13.09±0.58
CA	MAE	**19.95±0.42**	20.06±0.36	20.61±0.44	20.52±0.51	20.64±0.48	20.86±0.46	20.48±0.69	20.40±0.43
	RMSE	**32.67±0.53**	32.99±1.08	33.24±0.75	33.81±0.73	34.04±1.34	34.04±0.70	33.46±0.99	33.30±1.57
	MAPE	**14.21±0.24**	15.61±0.52	14.58±0.34	14.61±0.33	14.63±0.79	14.76±0.13	14.24±0.38	14.32±0.26
KnowAir	MAE	**20.27±0.51**	20.81±0.38	20.56±0.73	21.02±0.61	21.31±0.92	21.46±0.63	20.71±0.97	20.51±0.46
	RMSE	**29.75±0.54**	30.08±0.52	30.19±0.79	30.48±0.54	30.74±0.42	30.86±0.39	31.07±0.51	29.85±0.48
	MAPE	**47.29±1.41**	59.44±2.68	51.49±2.16	57.48±1.69	62.85±3.16	63.14±3.76	57.29±2.34	57.24±2.41

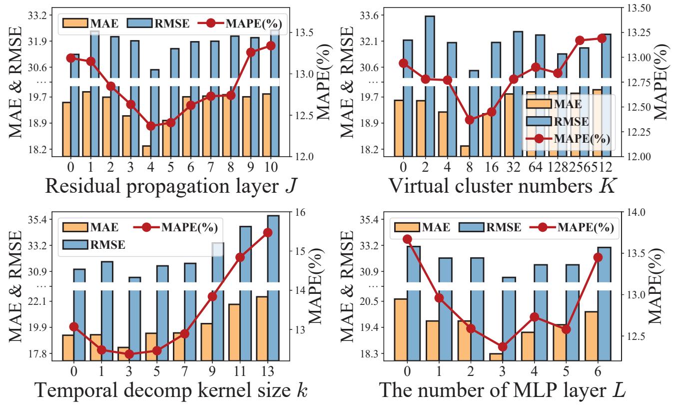


Figure 5: Hyperparameter sensitivity experiment.

Virtual cluster numbers K**.** A moderate number of virtual nodes, which corresponds to the core numbers, can effectively capture global spatiotemporal information. On the other hand, an excessive number of virtual nodes increases complexity without performance improvement and may even absorb excessive environmental noise, leading to performance decline.

Temporal decomposition kernel size k**.** Smaller kernel sizes exhibit similar and good stability in feature extraction. However, as the size increases, performance rapidly deteriorates due to the loss of local temporal information.

The number of MLP layers L**.** When we use 3 MLP layers for spatiotemporal learning (in Equation 9) and residual modeling (in Equation 16), BiST can achieve optimal prediction performance. This is because, with fewer layers, the model may fail to capture complex spatiotemporal correlations, leading to underfitting. Conversely, with too many layers, the increased model complexity can make learning more difficult, often resulting in overfitting.

5.7 Spatiotemporal Deviation Modeling (Q.6)

We evaluate the effectiveness of the model in handling spatiotemporal inconsistencies. Using the SD dataset as an example, we calculate the percentage change between the average values of the input data and the label data. We study two scenarios: a sudden increase and a sharp decrease in label data.

As shown in Table 8 and Figure 6, relative to existing state-of-the-art spatiotemporal prediction models, BiST can more effectively handle spatiotemporal inconsistent data. While STID claims to use the node embedding method to handle such data, existing models only involve a forward spatiotemporal learning process without utilizing label information, and hence they are unable to effectively deal with complex inconsistencies. In contrast, BiST includes a backward process that leverages label information to help the model better eliminate such disparities.

Table 8: Performance comparisons under severe cases on SD datasets. "Surge" refers to the growth multiple of the mean value of future data, while "Plummet" signifies the percentage decrease in the mean value of future data. "×": multiple.

Surge	1× ∼ 10×		10× ∼ 100×		100× ∼ 1000×	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
STGCN	7.82±0.18	20.61±0.32	13.75±0.25	26.36±0.28	18.19±0.11	36.45±0.47
STID	5.84±0.09	15.69±0.11	12.75±0.13	27.83±0.17	13.69±0.06	26.05±0.24
STAEformer	7.59±0.11	15.97±0.32	15.12±0.19	28.53±0.22	12.66±0.29	20.41±0.21
D^2STGNN	5.68±0.05	15.88±0.15	10.81±0.11	21.86±0.29	16.03±0.07	26.99±0.35
Ours	**5.33±0.07**	**14.50±0.14**	**10.41±0.12**	**20.29±0.23**	**11.53±0.14**	**18.67±0.18**

Plummet	25% ∼ 50%		50% ∼ 75%		75% ∼ 100%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
STGCN	13.51±0.25	27.86±0.37	18.45±0.24	28.49±0.39	23.84±0.48	36.43±0.48
STID	14.81±0.16	30.28±0.13	21.28±0.17	33.27±0.20	23.14±0.39	35.97±0.31
STAEformer	13.77±0.64	28.47±1.28	18.02±0.28	27.55±0.36	23.95±0.56	37.46±0.34
D^2STGNN	14.89±0.38	27.12±0.16	17.16±0.33	26.34±0.21	21.98±0.51	34.56±0.71
Ours	**12.99±0.18**	**26.04±0.24**	**16.75±0.32**	**25.32±0.26**	**21.29±0.45**	**33.24±0.52**

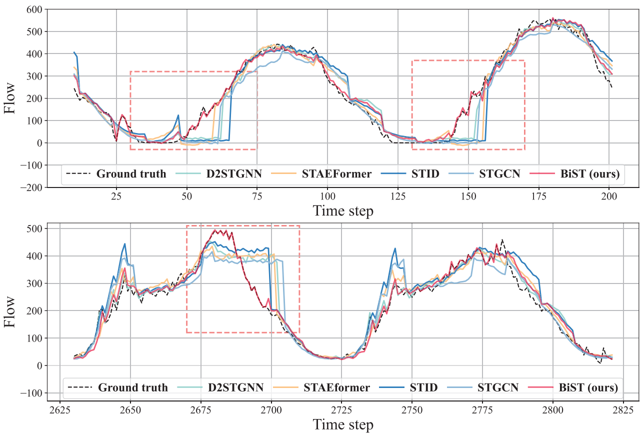


Figure 6: Prediction case visualization. The above figure illustrates the predictive performance of the model in the event of a surge of data, while the lower figure depicts the model's performance in the context of a sudden decline in data flow.

5.8 Modeling Multi-step Temporal Dependency for Residual Learning (Q.7)

After obtaining the residual representation $\mathbf{Z}_\mathrm{R}$, we employ two methods: LSTM and Transformer, to explicitly model dependencies between different time steps in $\mathbf{Z}_\mathrm{R}$. These variants are defined as Ours-LSTM and Ours-Transformer. As shown in Table 9, the results demonstrate that these alternative variants exhibit inferior performance compared to our model. The potential reason is that complex backward residual modeling networks might overly emphasize residuals while diminishing the effectiveness of spatiotemporal feature learning, reducing its effectiveness.

Table 9: Ablation experiment on modeling time step length.

Dataset	Avg.	**Ours**	Ours-LSTM	Ours-Transformer
SD	MAE	**18.30±0.37**	19.52±0.35	19.63±0.42
	RMSE	**30.44±0.42**	31.84±0.43	32.22±0.54
	MAPE	**12.37±0.31**	12.73±0.23	12.64±0.25
CA	MAE	**19.95±0.42**	20.41±0.49	Out-of-Memory
	RMSE	**32.67±0.53**	33.18±0.43	
	MAPE	**14.21±0.24**	14.48±0.38	
KnowAir	MAE	**20.27±0.51**	20.34±0.52	21.48±0.54
	RMSE	**29.75±0.54**	29.95±0.56	32.48±0.56
	MAPE	**47.29±1.41**	48.12±1.53	52.47±1.51

5.9 Case Study (Q.8)

5.9.1 Interpreting BiST prediction. The SHAP (**SH**apley **A**dditive ex**P**lanations) value [37, 38] represents a comprehensive measure of the importance of data features. It quantifies the average contribution of each feature to the predicted output, taking into account all possible combinations of feature perturbations. Following the STL decomposition [4], our BiST can be seen as a generalized additive model (GAM) [18],

$$\mathrm{Y} = \hat{\mathrm{Y}} + \mathrm{Y}_{\mathrm{err}}, \tag{25}$$

$$= \mathrm{Y}_{\mathrm{base}} + \mathrm{Y}_{\mathrm{cor}} + \mathrm{Y}_{\mathrm{err}}, \tag{26}$$

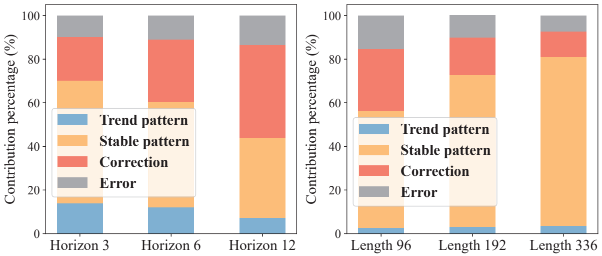
$$= \mathcal{F}\left(\mathrm{MLP}_1\left(\mathbf{X}_l\right) + \mathrm{MLP}_2\left(\mathbf{X}_S\right)\right) + \mathrm{Y}_{\mathrm{cor}} + \mathrm{Y}_{\mathrm{err}}. \tag{27}$$

where $\mathbf{X}_l$ and $\mathbf{X}_S$ represent the stable patterns and trend patterns of the time series in the forward spatiotemporal learning process, respectively. By averaging across all nodes, we calculate the SHAP values of the four components at 3 kinds of horizon steps. As shown in Figure 7, the application of SHAP values in BiST for spatiotemporal data reveals that stable temporal patterns play a crucial role in both short-term predictions (as illustrated in subplot (a)) and long-term predictions (as illustrated in subplot (b)). However, the influence of trend patterns gradually diminishes as the prediction horizon increases.

In short-term prediction, the slight increase in the "error" component stems from growing inconsistencies between input data and label information as prediction steps advance, making prediction more complex and potentially leading to decreased model accuracy. Consequently, the backward correction module plays an increasingly significant role by modeling inconsistent features to adjust baseline predictions. In long-term prediction, the contribution of the correction term remains substantial and cannot be overlooked.

5.9.2 Prompt embedding visualizations. In this section, we extract the trained embedding vectors from BiST for visualization to evaluate their effectiveness. As shown in Figure 8, we demonstrate the spatiotemporal prompt embeddings, the learnable embeddings of nodes and virtual nodes in the residual decomposition layer, as well as the receptive fields of virtual nodes and their visual position in real-world scenarios on SD dataset.

Temporal prompt embedding. Figure 8 (a) displays the visualization results of temporal prompt embedding $\boldsymbol{e}_T$ and $\boldsymbol{e}_D$. We observed that the prompts are precisely aggregated into 7 clusters with clear boundaries based on the day of the week. Moreover, the parts representing the time of day within each cluster exhibit distinct patterns, aggregating into smaller groups, demonstrating that the temporal prompts can provide clear temporal side information.

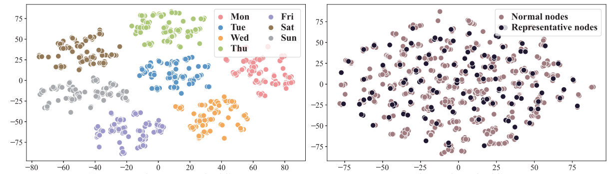


(a) Short-term prediciton (b) Long-term prediciton

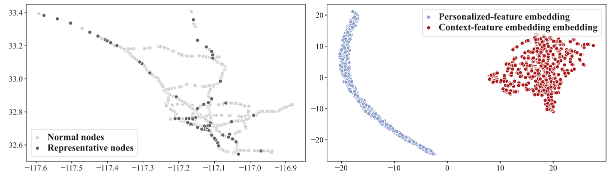
Figure 7: The SHAP values of decomposed components of BiST in SD dataset and (Daily) XXLTraffic dataset.

Node embedding visualization. Using SD dataset, we first extract the hierarchical receptive field S in the Equation 11. We select nodes with higher correlation in each cluster, which are denoted as 'representative nodes'. The feature of these nodes are close to those of clusters. The remaining nodes are called 'normal nodes'. Figure 8 (b) presents the node embedding visualization of these nodes $\boldsymbol{e}_S$, which indicates that the node embeddings are clustered, effectively learning hierarchical information. At the same time, our method successfully extracts shared features as the representative nodes are positioned near the center of each cluster. We further illustrate the distribution of these nodes in the real-world road network, as shown in Figure 8 (c).

Personalized-feature and context-feature embeddings. Figure 8 (d) visualizes the personalized-feature E_q and context-featur embeddings E_k. The context features which are shared among the nodes exhibit a clustered distribution, while the personalized features of each node show a strip-like distribution.



(a) Temporal prompt embedding. (b) Node embedding.



(c) Distribution of Nodes in road network locations. (d) Personalized-feature and context-featur embedding.

Figure 8: Visualization of various embeddings.

6 CONCLUSION

This paper presents a lightweight spatiotemporal prediction model based on MLP, achieving competitive predictive performance while maintaining low computational complexity and memory usage. The model effectively addresses inconsistencies between the label and the input information, thereby enhancing the overall performance. We propose a novel spatiotemporal decoupling module for capturing residuals, which decomposes spatiotemporal features into node-shared contextual features and node-specific features. Across more than a dozen datasets, we demonstrate the model's competitive accuracy, high training efficiency, and minimal memory overhead.

ACKNOWLEDGMENTS

This paper is supported by the National Natural Science Foundation of China (No.62072427, No.12227901), the Project of Stable Support for Youth Team in Basic Research Field, CAS(No.YSBR005), and Academic Leaders Cultivation Program, USTC.

A MATHEMATICAL PROOFS

In this section, we provide the proof of Theorem 1. Leveraging the framework of Gaussian Markov random fields [44], our proof unfolds by initially demonstrating the soundness of the foundational prediction term through the conditional distribution. Subsequently, we advance this groundwork by introducing supplementary conditional constraints to substantiate the veracity of Theorem 1.

Proof of Theorem 1 By the definition of Gaussian Markov random fields (GMRF) [11], we can define the multivariate Gaussian distribution of probability density function corresponding to the joint spatiotemporal variable $\boldsymbol{T} = [\boldsymbol{X}, \boldsymbol{Y}] \in \mathbb{R}^{(T+T_P)\times N \times c}$,

$$f_{\boldsymbol{T}}(\mathbf{T}) = (2\pi)^{\frac{-N(T+T_P)}{2}} \det\left(\boldsymbol{\Gamma}^{-1}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2}\operatorname{vec}(\mathbf{T})^{\top}\boldsymbol{\Gamma}\operatorname{vec}(\mathbf{T})\right), \tag{28}$$

where $\boldsymbol{\Gamma} = \Sigma^{-1} \in \mathbb{R}^{[(T+T_P)N]\times[(T+T_P)N]}$ is the precision matrix, i.e., the inverse of covariance matrix Σ. $\boldsymbol{\Gamma}$ represents the dependency between $(T+T_P)\times N$ variables in GMRF, and $\operatorname{vec}(\cdot)$ is the vectorization operator of tensor. Here $\boldsymbol{\Gamma}$ reflects the dependence of variables in the GMRF, which can be computed as

$$\boldsymbol{\Gamma} \coloneqq (W \otimes \mathbf{I}_N) + \operatorname{diag}(h) \otimes \mathcal{A}(\mathbf{A}), \tag{29}$$

where $W \in \mathbb{R}^{(T+T_P)\times(T+T_P)}$ satisfying the symmetric positive definite and $\theta \in \mathbb{R}^{T+T_P}$ satisfying the entry positive are the pseudo parameters of the standard GMRF model. Detailed explanation of these parameters suggests a reference to [22]. $\operatorname{diag}(\cdot)$ is the diagonalization operator and $\otimes$ is the Kronecker product.

Spatiotemporal variables can be assumed as a multivariate Gaussian distribution, i.e., $\operatorname{vec}(\boldsymbol{T}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}^{-1})$. Without loss of generality, we divide the node set of the spatiotemporal graph into two disjoint unions to simplify subsequent calculations: $\mathcal{V} = V_1 \cup V_2$, i.e., $V_1 \cap V_2 = \emptyset$. Recall the conditional distribution of $\boldsymbol{Y}$ corresponding to the input $\mathbf{X}$ with its variable $\boldsymbol{X}$, we can get

$$\boldsymbol{Y}|\mathbf{X} \sim \mathcal{N}\left(\mathbb{E}\left[\boldsymbol{Y}|\mathbf{X}\right], \boldsymbol{\Gamma}_{\boldsymbol{Y}\boldsymbol{Y}}^{-1}\right), \tag{30}$$

where $\boldsymbol{\Gamma}_{\boldsymbol{Y}\boldsymbol{Y}}^{-1} \in \mathbb{R}^{(T_P N)\times(T_P N)}$ indicates the dependencies between the variables contained in label $\boldsymbol{Y}$. Hence the conditional distribution of $\mathbf{Y}_{t,V_1}$ respect to $\mathbf{Y}_{t,V_2}$ and $\mathbf{X}$ for the disjoint union $V_1 \cup V_2$ of node set V and arbitrary $t = \{1, 2, \ldots, T_P\}$ is

$$\begin{aligned}&\mathbf{Y}_{t,V_1}|\mathbf{X}, \mathbf{Y}_{t,V_2} \\ &\sim \mathcal{N}\left(\mathbb{E}\left[\mathbf{Y}_{t,V_1}|\mathbf{X}\right] + \boldsymbol{\Gamma}_{t,V_1V_1}^{-1}\boldsymbol{\Gamma}_{t,V_1V_2} \times_2 \left(\mathbb{E}\left[\mathbf{Y}_{t,V_1}|\mathbf{X}\right] - \mathbf{Y}_{t,V_2}\right), \boldsymbol{\Gamma}_{t,V_1V_1}^{-1}\right),\end{aligned} \tag{31}$$

where $\mathbf{Y}_{t,V_i} \coloneqq \left[\mathbf{Y}_{t,u,:}^{\top} \mid \forall u \in V_i\right]^{\top}$ for $i = \{1, 2\}$. Hence the above expectation equation is,

$$\begin{aligned}&\mathbb{E}\left[\mathbf{Y}_{t,V_1}|\mathbf{X}, \mathbf{Y}_{t,V_2}\right] \\ &= \mathbb{E}\left[\mathbf{Y}_{t,V_1}|\mathbf{X}\right] + \boldsymbol{\Gamma}_{t,V_1V_1}^{-1}\boldsymbol{\Gamma}_{t,V_1V_2}\left(\mathbb{E}\left[\mathbf{Y}_{t,V_1}|\mathbf{X}\right] - \mathbf{Y}_{t,V_2}\right), \\ &= \mathbb{E}\left[\mathbf{Y}_{t,V_1}|\mathbf{X}\right] + \left(\mathbf{I}_N + \alpha_t\mathcal{A}(\mathbf{A})\right)_{V_1V_1}^{-1}\left(\mathbf{I}_N + \alpha_t\mathcal{A}(\mathbf{A})\right)_{V_1V_2} \times_2 \mathbf{c}_{t,V_2},\end{aligned} \tag{32}$$

where $\alpha_t = \frac{h_t}{W_{T+t,T+t}}$ and $\times_2$ means the multiplication operation of the second dimension of two matrices. The term $\left(\mathbf{I}_N + \alpha_t\mathcal{A}(\mathbf{A})\right)_{V_1,V_2}$ indicates the submatrix consisting of rows corresponding to entries in V_1 and columns corresponding to entries in V_2 for $\mathbf{I}_N + \alpha_t\mathcal{A}(\mathbf{A})$, which illustrates the dynamics of residual propagation in this context. Based on the expansion of Neumann series [41], the above equation can expand as follows,

$$\begin{aligned}&\mathbb{E}\left[\mathbf{Y}_{t,V_1}|\mathbf{X}, \mathbf{Y}_{t,V_2}\right] = \mathbb{E}\left[\mathbf{Y}_{t,V_1}|\mathbf{X}\right] \\ &\quad + (1-\gamma_t)\sum_{k=0}^{\infty}\left(\gamma_t\tilde{\mathbf{A}}_{V_1,V_1}\right)^k \left(\mathbf{I}_N + \alpha_t\mathcal{A}(\mathbf{A})\right)_{V_1,V_2} \times_2 \mathbf{c}_{t,V_2},\end{aligned} \tag{33}$$

where $\gamma_t = \alpha_t/(1+\alpha_t)$. $\tilde{\mathbf{A}}_{V_1,V_1}$ indicates the submatrix consisting of rows and columns corresponding to entries in V_1 for $\tilde{\mathbf{A}}$. It must be noted, however, that the results of the closed form are independent of the node disjoint union partition chosen, as determined by the equivariance of the GMRF [2]. Hence, the case we considered in the Theorem 1 is just a special example in the proof when $V_1 = \{u\}$ and $V_2 = \mathcal{V} \setminus \{u\}$. Proof of completion.□

B ADDITIONAL EXPERIMENTS

In the study of LargeST [33], the authors employed 2019 data as a case study to compare the predictive performance of different models. To enable a straightforward comparison, we also report the average performance metrics across 12 time steps from the 2019 data. As illustrated in Table 10, BiST consistently outperforms all baseline models across the four datasets, with 75% of the performance metrics showing a relative improvement of over 5%. This further validates the superiority of the BiST model in handling large-scale spatiotemporal data.

Table 10: Short-term performance comparisons in LargeST (2019). The unit of MAPE is percent (%).

Dataset	SD 2019			GBA 2019			GLA 2019			CA 2019		
Method	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	19.89±0.51	33.84±0.25	13.96±0.12	23.92±0.33	39.41±0.45	18.54±0.21	22.66±0.47	38.78±0.51	14.18±0.14	21.64±0.25	36.25±0.82	16.64±0.27
GWNET	18.07±0.42	29.67±0.21	11.69±0.19	20.92±0.31	33.47±0.48	17.96±0.29	21.32±0.36	33.56±0.31	13.26±0.12	22.21±0.22	34.05±0.71	17.63±0.21
STNorm	19.19±0.75	31.56±0.31	12.16±0.18	22.32±0.22	35.73±0.57	17.09±0.19	22.11±0.73	35.12±0.97	13.42±0.69	20.24±0.25	33.51±0.78	14.75±0.36
STID	18.44±0.14	32.05±0.37	12.52±0.14	20.93±0.18	35.31±0.38	17.65±0.05	20.77±0.09	35.04±0.23	13.35±0.13	19.21±0.19	31.69±0.18	15.19±0.07
STGODE	19.34±0.68	34.29±1.19	13.67±0.28	21.94±0.37	35.97±0.84	18.52±0.26	21.69±0.36	35.98±0.54	13.75±0.22	21.05±0.26	36.85±0.45	17.02±0.28
ASTGCN	23.55±1.83	39.45±3.31	16.54±1.32	26.89±0.98	41.76±2.26	24.23±1.17	27.86±1.07	44.84±2.32	16.85±1.12	Out of Memory		
AGCRN	18.42±0.38	32.62±0.47	13.36±0.29	21.51±0.27	34.38±0.87	17.01±0.33	20.39±0.36	34.73±1.17	12.74±0.25			
DSTAGNN	21.87±0.56	30.91±1.54	12.98±1.19	24.23±1.18	37.29±1.15	20.49±0.95	Out of Memory					
STAEformer	18.96±0.42	31.79±0.78	13.23±0.31	21.79±0.56	35.12±1.30	17.07±0.27						
DGCRN	18.07±0.24	30.19±0.48	12.14±0.23	21.47±0.46	33.99±0.42	17.15±0.31						
D^{2}STGNN	17.81±0.21	29.72±0.49	11.74±0.36	20.92±0.25	33.98±0.46	15.08±0.13						
Ours	**16.19±0.17**	**27.72±0.33**	**10.59±0.12**	**19.31±0.15**	**32.39±0.31**	**14.43±0.07**	**19.14±0.12**	**31.81±0.29**	**11.33±0.07**	**17.58±0.18**	**31.01±0.22**	**12.79±0.09**

REFERENCES

[1] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems* 33 (2020), 17804–17815.

[2] Juan Baz, Irene Díaz, Susana Montes, and Raúl Pérez-Fernández. 2022. Some results on the Gaussian Markov Random Field construction problem based on the use of invariant subgraphs. *Test* 31, 3 (2022), 856–874.

[3] Kyunghyun Cho. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[4] Robert B Cleveland, William S Cleveland, Jean E McRae, Irma Terpenning, et al. 1990. STL: A seasonal-trend decomposition. *J. off. Stat* 6, 1 (1990), 3–73.

[5] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. 2021. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 269–278.

[6] Dazhao Du, Bing Su, and Zhewei Wei. 2023. Preformer: predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[7] Yuchen Fang, Fang Zhao, Yanjun Qin, Haiyong Luo, and Chenxing Wang. 2022. Learning all dynamics: Traffic forecasting via locality-aware spatio-temporal joint transformer. *IEEE Transactions on Intelligent Transportation Systems* 23, 12 (2022), 23433–23446.

[8] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 364–373.

[9] Cyril Furtlehner, Jean-Marc Lasgouttes, Alessandro Attanasi, Marco Pezzulla, and Guido Gentile. 2021. Short-term forecasting of urban traffic using spatio-temporal Markov field. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021), 10858–10867.

[10] Haotian Gao, Renhe Jiang, Zheng Dong, Jinliang Deng, Yuxin Ma, and Xuan Song. 2024. Spatial-Temporal-Decoupled Masked Pre-training for Spatiotemporal Forecasting. arXiv:2312.00516 [cs.LG] https://arxiv.org/abs/2312.00516

[11] Nathaniel R Goodman. 1963. Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction). *The Annals of mathematical statistics* 34, 1 (1963), 152–177.

[12] Xiaochuan Gou, Ziyue Li, Tian Lan, Junpeng Lin, Zhishuai Li, Bingyu Zhao, Chen Zhang, Di Wang, and Xiangliang Zhang. 2024. XTraffic: A Dataset Where Traffic Meets Incidents with Explainability and More. *arXiv preprint arXiv:2407.11477* (2024).

[13] Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* (2023).

[14] Renxiang Guan, Zihao Li, Wenxuan Tu, Jun Wang, Yue Liu, Xianju Li, Chang Tang, and Ruyi Feng. 2024. Contrastive Multiview Subspace Clustering of Hyperspectral Images Based on Graph Convolutional Networks. *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), 1–14.

[15] Renxiang Guan, Wenxuan Tu, Zihao Li, Hao Yu, Dayu Hu, Yuzeng Chen, Chang Tang, Qiangqiang Yuan, and Xinwang Liu. 2024. Spatial-Spectral Graph Contrastive Clustering with Hard Sample Mining for Hyperspectral Images. *IEEE Transactions on Geoscience and Remote Sensing* (2024), 1–16.

[16] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 922–929.

[17] Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. 2024. SOFTS: Efficient Multivariate Time Series Forecasting with Series-Core Fusion. *arXiv preprint arXiv:2404.14197* (2024).

[18] Trevor J Hastie. 2017. Generalized additive models. In *Statistical models in S*. Routledge, 249–307.

[19] Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. 2024. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 12608–12616.

[20] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. 2023. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems* 36 (2023), 46885–46902.

[21] Guangyu Huo, Yong Zhang, Boyue Wang, Junbin Gao, Yongli Hu, and Baocai Yin. 2023. Hierarchical spatio–temporal graph convolutional networks and transformer network for traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems* 24, 4 (2023), 3855–3867.

[22] Junteng Jia and Austin R Benson. 2022. A unifying generative model for graph learning algorithms: Label propagation, graph convolutions, and combinations. *SIAM Journal on Mathematics of Data Science* 4, 1 (2022), 100–125.

[23] Yue Jiang, Xiucheng Li, Yile Chen, Shuai Liu, Weilong Kong, Antonis F Lentzakis, and Gao Cong. 2024. SAGDFN: A Scalable Adaptive Graph Diffusion Forecasting Network for Multivariate Time Series Forecasting. *arXiv preprint arXiv:2406.12282* (2024).

[24] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[25] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*. PMLR, 11906–11917.

[26] Yuen Hoi Lau and Raymond Chi-Wing Wong. 2021. Spatio-temporal graph convolutional networks for traffic forecasting: Spatial layers first or temporal layers first?. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 427–430.

[27] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. 2023. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data* 17, 1 (2023), 1–21.

[28] Xin Li, Yuhong Guo, and Dale Schuurmans. 2015. Semi-supervised zero-shot classification with label representation learning. In *Proceedings of the IEEE international conference on computer vision*. 4211–4219.

[29] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[30] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. 2015. Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*. 1–10.

[31] Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. 2024. SparseTSF: Modeling Long-term Time Series Forecasting with 1k Parameters. *arXiv preprint arXiv:2405.00946* (2024).

[32] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. 2023. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 4125–4129.

[33] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. 2024. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in Neural Information Processing Systems* 36 (2024).

[34] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* (2023).

[35] Zhipeng Liu, Peibo Duan, Xiaosha Xue, Changsheng Zhang, Wenwei Yue, and Bin Zhang. 2024. A dynamic hypergraph attention network: Capturing market-wide spatiotemporal dependencies for stock selection. *Applied Soft Computing* (2024), 112524.

[36] Jiecheng Lu, Xu Han, Yan Sun, and Shihao Yang. 2024. CATS: Enhancing Multivariate Time Series Forecasting by Constructing Auxiliary Time Series as Exogenous Variables. *arXiv preprint arXiv:2403.01673* (2024).

[37] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* (2018).

[38] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[39] Xiang Ma, Xuemei Li, Lexin Fang, Tianlong Zhao, and Caiming Zhang. 2024. U-Mixer: An Unet-Mixer Architecture with Stationarity Correction for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 14255–14262.

[40] Hao Miao, Yan Zhao, Chenjuan Guo, Bin Yang, Zheng Kai, Feiteng Huang, Jiandong Xie, and Christian S Jensen. 2024. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *ICDE*.

[41] Hervé Moulinec, Pierre Suquet, and Graeme W Milton. 2018. Convergence of iterative methods based on Neumann series for composite materials: Theory and practice. *Internat. J. Numer. Methods Engrg.* 114, 10 (2018), 1103–1130.

[42] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. [n.d.]. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.

[43] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. 2024. DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting. *arXiv preprint arXiv:2412.10859* (2024).

[44] Havard Rue and Leonhard Held. 2005. *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC.

[45] Gabi Shalev, Yossi Adi, and Joseph Keshet. 2018. Out-of-distribution detection using multiple semantic label representations. *Advances in Neural Information Processing Systems* 31 (2018).

[46] Zezhi Shao, Fei Wang, Zhao Zhang, Yuchen Fang, Guangyin Jin, and Yongjun Xu. 2023. Hutformer: Hierarchical u-net transformer for long-term traffic forecasting. *arXiv preprint arXiv:2307.14596* (2023).

[47] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. 2022. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4454–4458.

[48] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting.

In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 1567–1577.

[49] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S. Jensen. 2022. Decoupled Dynamic Spatial-Temporal Graph Neural Network for Traffic Forecasting. *Proc. VLDB Endow.* 15, 11 (2022), 2733–2746.

[50] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 914–921.

[51] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[52] Guancheng Wan, Wenke Huang, and Mang Ye. 2024. Federated graph learning under domain shift with generalizable prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 15429–15437.

[53] Binwu Wang, Jiaming Ma, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang. 2024. Stone: A spatio-temporal ood learning framework kills both spatial and temporal shifts. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2948–2959.

[54] Binwu Wang, Pengkun Wang, Yudong Zhang, Xu Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. 2024. Towards dynamic spatial-temporal graph learning: A decoupled perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9089–9097.

[55] Binwu Wang, Pengkun Wang, Yudong Zhang, Xu Wang, Zhengyang Zhou, and Yang Wang. 2024. Condition-guided urban traffic co-prediction with multiple sparse surveillance data. *IEEE Transactions on Vehicular Technology* (2024).

[56] Binwu Wang, Pengkun Wang, Zhengyang Zhou, Zhe Zhao, Wei Xu, and Yang Wang. 2024. Make Bricks with a Little Straw: Large-Scale Spatio-Temporal Graph Learning with Restricted GPU-Memory Capacity. *International Joint Conference on Artificial Intelligence* (2024).

[57] Binwu Wang, Yudong Zhang, Jiahao Shi, Pengkun Wang, Xu Wang, Lei Bai, and Yang Wang. 2023. Knowledge expansion and consolidation for continual traffic prediction with expanding graphs. *IEEE Transactions on Intelligent Transportation Systems* 24, 7 (2023), 7190–7201.

[58] Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. 2023. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2223–2232.

[59] Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. 2024. TimeMixer++: A General Time Series Pattern Machine for Universal Predictive Analysis. *arXiv preprint arXiv:2410.16032* (2024).

[60] Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. 2020. Pm2. 5-gnn: A domain knowledge enhanced graph neural network for pm2. 5 forecasting. In *Proceedings of the 28th international conference on advances in geographic information systems*. 163–166.

[61] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616* (2024).

[62] Wenchao Weng, Jin Fan, Huifeng Wu, Yujie Hu, Hao Tian, Fu Zhu, and Jia Wu. 2023. A decomposition dynamic graph convolutional recurrent network for traffic forecasting. *Pattern Recognition* 142 (2023), 109670.

[63] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* 34 (2021), 22419–22430.

[64] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121* (2019).

[65] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. 2020. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908* (2020).

[66] Du Yin, Hao Xue, Arian Prabowo, Shuang Ao, and Flora Salim. 2024. XXLTraffic: Expanding and Extremely Long Traffic Dataset for Ultra-Dynamic Forecasting Challenges. *arXiv preprint arXiv:2406.12693* (2024).

[67] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).

[68] Chengqing Yu, Fei Wang, Zezhi Shao, Tao Sun, Lin Wu, and Yongjun Xu. 2023. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 3062–3072.

[69] Chenyang Yu, Xinpeng Xie, Yan Huang, and Chenxi Qiu. 2024. Harnessing llms for cross-city od flow prediction. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*. 384–395.

[70] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. 2024. UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction. *arXiv preprint arXiv:2402.11838* (2024).

[71] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11121–11128.

[72] Weijia Zhang, Jindong Han, Zhao Xu, Hang Ni, Hao Liu, and Hui Xiong. 2024. Towards Urban General Intelligence: A Review and Outlook of Urban Foundation Models. *arXiv preprint arXiv:2402.01749* (2024).

[73] Yudong Zhang, Pengkun Wang, Binwu Wang, Xu Wang, Zhe Zhao, Zhengyang Zhou, Lei Bai, and Yang Wang. 2024. Adaptive and Interactive Multi-Level Spatio-Temporal Network for Traffic Forecasting. *IEEE Transactions on Intelligent Transportation Systems* (2024).

[74] Kai Zhao, Chenjuan Guo, Yunyao Cheng, Peng Han, Miao Zhang, and Bin Yang. 2023. Multiple time series forecasting with dynamic graph modeling. *Proceedings of the VLDB Endowment* 17, 4 (2023), 753–765.

[75] Zuduo Zheng and Dongcai Su. 2016. Traffic state estimation through compressed sensing and Markov random field. *Transportation Research Part B: Methodological* 91 (2016), 525–554.

[76] Yang Zhou and Yan Huang. 2018. Context aware flow prediction of bike sharing systems. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2393–2402.

[77] Zhengyang Zhou, Qihe Huang, Binwu Wang, Jianpeng Hou, Kuo Yang, Yuxuan Liang, and Yang Wang. 2024. ComS2T: A complementary spatiotemporal learning system for data-adaptive model evolution. *arXiv preprint arXiv:2403.01738* (2024).

[78] Ziyun Zou, Yinghui Jiang, Lian Shen, Juan Liu, and Xiangrong Liu. 2025. LOHA: Direct Graph Spectral Contrastive Learning Between Low-pass and High-pass Views. *arXiv preprint arXiv:2501.02969* (2025).