Learning Gaussian Mixture Representations for Tensor Time Series Forecasting

Jiewen Deng[1] , **Jinliang Deng**[1,2] , **Renhe Jiang**[3,1]* and **Xuan Song**[1,3]
[1]Southern University of Science and Technology
[2]University of Technology Sydney
[3]The University of Tokyo
dengjw1@outlook.com, jinliang.deng@student.uts.edu.au
jiangrh@csis.u-tokyo.ac.jp, songx@sustech.edu.cn

Abstract

Tensor time series (TTS) data, a generalization of one-dimensional time series on a high-dimensional space, is ubiquitous in real-world scenarios, especially in monitoring systems involving multi-source spatio-temporal data (e.g., transportation demands and air pollutants). Compared to modeling time series or multivariate time series, which has received much attention and achieved tremendous progress in recent years, tensor time series has been paid less effort. Properly coping with the tensor time series is a much more challenging task, due to its high-dimensional and complex inner structure. In this paper, we develop a novel TTS forecasting framework, which seeks to individually model each heterogeneity component implied in the time, the location, and the source variables. We name this framework as **GMRL**, short for Gaussian Mixture Representation Learning. Experiment results on two real-world TTS datasets verify the superiority of our approach compared with the state-of-the-art baselines. Code and data are published on https://github.com/beginner-sketch/GMRL.

1 Introduction

Tensor time series (TTS) data occur as sequences of multi-dimensional arrays [Rogers *et al.*, 2013; Jing *et al.*, 2021]. For example, urban public transportation management involves data coming from multiple times, locations, and transportation modes, i.e., Taxi/Bike Inflow/Outflow [Zhang *et al.*, 2016; Zhang *et al.*, 2017], based on which compelling predictions are crucial to subsequent decision-making, especially for those urban-sensitive cases. Compared with multivariate time series (MTS), TTS involves an additional variable apart from the initial time and location variables, namely the *source* variable. The most significant difference between MTS and TTS is that MTS involves data in a matrix, where the rows represent the time variable and the columns define the location variable. In contrast, TTS concerns tensor data, which usually takes on more substantial heterogeneity and dynamics. Although identifying and extracting similar patterns is reasonable and typical for time series prediction, the components over variables can be dynamic in many real-world applications. Failure to explicitly obtain the quantitative form of patterns may mistakenly cause the model to capture the pattern change in TTS. We use an example of NYC traffic demand data in Figure 1 to illustrate it. We collect the historical data of the three frames/locations/sources and plot their data distribution in the upper part of Figure 1. Based on the characteristics (the mean and variance) of the data distributions, we can trivially draw a conclusion that the instance at 6 am is more similar to 12 am than 8 pm. This is consistent with the evolving patterns of observation in the lower part of Figure 1. Such distinguishability over the *time* variable also holds water over the *location* and *source* variables. Given the complex patterns, capturing heterogeneities arising from multiple variables is an essential problem in TTS forecasting.

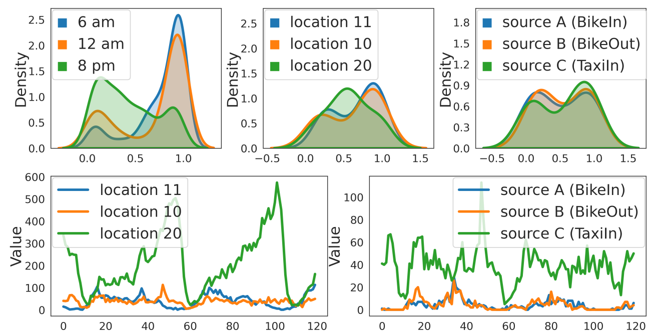

Figure 1: Demonstration of heterogeneous NYC traffic demand data. It contains four sources: Bike Inflow, Bike Outflow, Taxi Inflow, and Taxi Outflow. Regarding probability density (upper), dynamic heterogeneity exists in the time/location/source variables, consistent with the evolution trend (lower).

Deep forecasting models have achieved great success in more effective time series forecasting. Graph neural network (GNN) [Wu *et al.*, 2020b; Shao *et al.*, 2022], temporal convolution networks (TCN) [Wu *et al.*, 2020b], recurrent neural networks (RNN) [Lai *et al.*, 2018], and attention mechanisms [Wu *et al.*, 2021; Zheng *et al.*, 2020] are well-used for prediction problems. Each individual or a combination of these manipulations performs well in modeling the characteristics of a single domain or spatial-temporal regularities. Such methods are more prone to focus on low-level hi-

*Corresponding Author

erarchical information and are blind to the rich information and challenges brought by source variables. An increasing number of recent works are drawing forth the evolution from other modes as an auxiliary input, adding ancillary information to borrow similar historical patterns [Wu *et al.*, 2020a; Fang *et al.*, 2021; Han *et al.*, 2021]. These practices, to a certain degree, bring mode-wise awareness. Nevertheless, they mainly help with the spatial-temporal parts, and the heterogeneity or the interactions brought by variables (i.e., time, location, and source) are still underexplored. Recently, cluster-wise learning breaks the independence between instances by exploiting the latent cluster information among samples. The learned representations are expected to retain higher-level heterogeneous information by taking advantage of additional prior information brought by clustering. However, existing approaches [Zhu *et al.*, 2021; Duan *et al.*, 2022] (i) neglect the dynamic nature of time series, assuming that both the representations and the members of the clusters will remain the same over time and (ii) the interactions over multiple variables as well, which are not applicable for intricate and complex TTS. Thus far, while evolution regularities in time series have been studied intensively, dynamic heterogeneity and interactions of multiple variables in TTS have yet to be adequately tackled.

Therefore, we are motivated to propose a novel TTS learning framework for accurate forecasting, namely Gaussian Mixture Representation Learning (**GMRL**). It consists of three steps: (i) incorporating the multiple variables information into the model through a TTS embedding (*TTSE*); (ii) designing a dynamic Gaussian mixture representation extractor to explicitly represent the complicated heterogeneous components among multiple variables; (iii) leveraging techniques of memory that enhances the learned representation to learn to distinguish and generalize to diverse scenarios. The contributions of this work are summarized as follows:

- We design a Gaussian Mixture Representation Extractor (GMRE), to explicitly disentangle and quantify the heterogeneity components in the time, the location, and the source variables.
- We propose a Hidden Representation Augmenter (HRA) to equip the learned representations with sequence-specific global patterns on the fly, intrinsically improving model adaptability to TTS.
- We conduct thorough experiments on two real-world datasets, one is traffic dataset containing four sources of transportation demands (i.e., Taxi/Bike Inflow/Outflow), another is air quality dataset containing three sources of pollutants (i.e., PM2.5, PM10, SO_2). The results verify the superiority of our method over the state-of-the-arts.

2 Preliminaries

In this section, we formulate the tensor time series forecasting problem, and introduce the used notations (in Table 1).

Problem 1 (Tensor Time Series Forecasting). Given T consecutive time steps, L discretized locations, and S sources, the multi-source spatio-temporal data can be aggregated into a *Tensor Time Series* $X \in \mathbb{R}^{T\times L\times S}$. Given X, the tensor time series forecasting aims to forecast the next O steps as $\hat{Y} \in \mathbb{R}^{O\times L\times S}$, which can be expressed by the following conditional distribution:

Notation	Description
L / S / T	Number of locations/sources/time steps.
O	Number of horizons.
K	Number of cluster components.
$X \in \mathbb{R}^{T\times L\times S}$ / $\hat{Y} \in \mathbb{R}^{O\times L\times S}$	Input/Output tensor.
$E \in \mathbb{R}^{T\times L\times S\times d_z}$	Tensor time series embedding.
$H \in \mathbb{R}^{T\times L\times S\times d_k}$	Initial representation.
$\hat{H} \in \mathbb{R}^{T\times L\times S\times d_k}$	Representation from GMRE.
$H^{(\mathrm{gm})} \in \mathbb{R}^{T\times L\times S\times 2d_k}$	Gaussian mixture representation.
$H^{(\mathrm{te})} \in \mathbb{R}^{T\times L\times S\times d_k}$	Representation from TE.
$H^{(\mathrm{sc})} \in \mathbb{R}^{L\times S\times d_k}$	Representation after skip connection.
$H^{(\mathrm{aug})} \in \mathbb{R}^{L\times S\times (d_k+d_m)}$	Representation augmented by HRA.

Table 1: Notations and Descriptions

$$P(\hat{Y}|X) = \prod_{t=1}^{O} P(\hat{Y}_{t,:,:}|X) \qquad (1)$$

3 Methodology

In this section, we elaborate on the technical details of the proposed framework Gaussian Mixture Representation Learning (GMRL), as demonstrated in Figure 2.

3.1 Tensor Time Series Embedding

Since the evolution patterns of TTS are constrained by the location and the source variables, it is crucial to incorporate the variables' information into the predictive model. Specifically, we generate *location embedding* and *source embedding* to encode corresponding variables into vectors that preserve static information, respectively expressed as $e_l^L \in \mathbb{R}^{d_z}$ and $e_s^S \in \mathbb{R}^{d_z}$. The above embedding vectors only provide static representations, which could not represent the dynamic correlations among locations and sources. Thus, we further utilize *temporal embedding* $e_t^T \in \mathbb{R}^{d_z}$ to encode each historical time step into a vector. To obtain the variable representations, we fuse the aforementioned embedding vectors in an additive manner. For location l and source s at time step t, the embedding is defined as: $e_{t,l,s} = e_t^T + e_l^L + e_s^S$. Then we get the TTSE ($E \in \mathbb{R}^{T\times L\times S\times d_z}$) of L locations and S sources in T time steps. Next, the initial representation fed into subsequent modules is $H = [f(X), E]$, where $f(\cdot)$ represents linear transformation and $[,]$ denotes concatenation operation.

3.2 Gaussian Mixture Representation Extractor

The canonical cluster analysis method is used to reveal the underlying nature and patterns of data, enabling the classification of the internal correlation structure of data. However, existing time series modeling methods rely on cluster distinctions that do not necessarily indicate an optimal solution. Additionally, time series data feature a dynamic context, implying that the characteristics of the clusters, which describe real-time time series data, will change over time as a function of the dynamic context.

To overcome the above limitations, we develop a dynamic Gaussian Mixture Representation Extractor (GMRE) to explicitly capture the patterns of time series data based on the

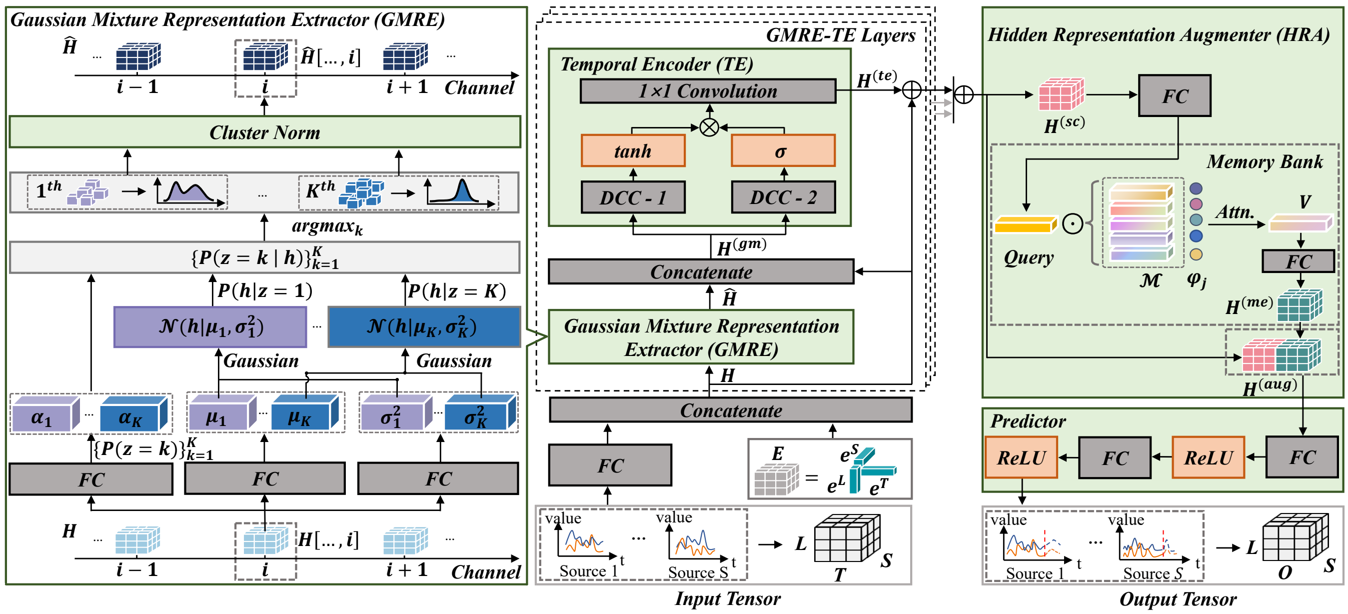

Figure 2: Gaussian Mixture Representation Learning (GMRL): (i) taking a three-dimension TTS as input; (ii) multiple proposed Gaussian Mixture Representation Extractors (GMRE) and Temporal Encoders (TE) are stacked layer by layer; (iii) applying the skip connection to fusion the representation output from each GMRE-TE layer; (iv) a Hidden Representation Augmenter (HRA) is deployed to strengthen representation with global patterns; (v) a Predictor is used to generate the prediction for TTS.

underlying context. Moreover, we expect different channels to hold information about distinct contextual attributes, i.e., the factors of variation, exhibiting mutual independence and complementarity. For instance, some channels correspond to spatial identities, remaining invariant over time, while others correspond to temporal identities, displaying variations, potentially with different frequencies, over time. Motivated by this, we perform channel-wise clustering instead of the conventional sample-wise clustering, in order to separate the clusters for each channel. A practical advantage of injecting inter-channel independence lies in reducing the number of learnable parameters involved in the model, as modeling interactions among the channels is no longer necessary.

Given a representation $H \in \mathbb{R}^{T\times L\times S\times d_k}$ derived from an input tensor X, GMRE aims to learn a mixture of Gaussian distributions over each scalar value within H. Technically, the probability distribution is conditioned on the observation (i.e., input tensor). However, for simplicity, we omit this condition in the following probability formulas. Firstly, we introduce a latent variable $z[i] \in [1, K]$ to identify the membership (belonging to which of the K clusters) of each scalar value for each channel $i \in [1, d_k]$. Given $h \in \mathbb{R}^{d_k}$ (one of the $T \times L \times S$ small cubes in Figure 2) from the representation H, GMRE infers the probability distribution for each scalar value in $h[i]$ (i.e., i-th channel of h). Formally, the probability of $h[i]$ can be decomposed as follows:

$$P(h[i]) = \sum_{k=1}^{K} P(z[i] = k)P(h[i]|z[i] = k) \tag{2}$$

Then, we can specify the expressions for $P(z[i] = k)$ and $P(h[i]|z[i] = k)$. Formally, we obtain the prior distribution of z as follows:

$$\begin{aligned} P(z[i] = k) &= \alpha_k(H[..., i]) \\ &= \frac{\exp(W^{\alpha}_{k,i}\ \mathrm{vec}(H[..., i]))}{\sum_{j=1}^{K}\exp(W^{\alpha}_{j,i}\ \mathrm{vec}(H[..., i]))} \end{aligned} \tag{3}$$

where the membership score associated with the i^{th} channel belonging to the k^{th} cluster is parameterized by a linear mapping with $W^{\alpha}_{k,i} \in \mathbb{R}^{TLS}$. For $P(h[i]|z[i] = k)$, we assume it follows a Gaussian distribution $\mathcal{N}(h[i]|\mu_{k,i}, \sigma^2_{k,i})$, where the mean $\mu_{k,i} \in \mathbb{R}$ and variance $\sigma^2_{k,i} \in \mathbb{R}$ are generated by:

$$\begin{cases} \mu_{k,i} = W^{\mu}_{k,i}\mathrm{vec}(H[..., i]) + b^{\mu}_{k,i}, \\ \sigma^2_{k,i} = \exp\left(W^{\sigma}_{k,i}\mathrm{vec}\left(H[..., i]\right) + b^{\sigma}_{k,i}\right), \end{cases} \tag{4}$$

where $W^{\mu}_{k,i}, W^{\sigma}_{k,i} \in \mathbb{R}^{TLS}$ are weights to project representation $H[..., i]$ to $\mu_{k,i}$ and $\sigma^2_{k,i}$, and $b^{\mu}_{k,i}, b^{\sigma}_{k,i} \in \mathbb{R}$ are the corresponding biases. $\exp(\cdot)$ ensures that $\sigma^2_{k,i}$ remains greater than zero.

With $P(z[i] = k)$ and $P(h|z[i] = k)$ available, we can derive the posterior probability $P(z[i] = k|h[i])$ as follows, using Bayes' theorem:

$$\begin{aligned} P(z[i] = k|h[i]) &= \frac{P(z[i] = k)P(h[i]|z[i] = k)}{P(h[i])}, \\ &= \frac{\alpha_k(H[..., i]) \cdot \mathcal{N}(h[i]|\mu_{k,i}, \sigma^2_{k,i})}{\sum_j \alpha_j(H[..., i]) \cdot \mathcal{N}(h[i]|\mu_{j,i}, \sigma^2_{j,i})} \end{aligned} \tag{5}$$

Following Eq. 5, we can assign the i^{th} channel of h to cluster $k = \arg\max_k P(z[i] = k|h[i])$.

Cluster Normalization. Inspired by the simplex and efficiency of normalization technology, we devise *Cluster Norm* to increase the distance between clusters while reducing the distance between samples within a cluster. Cluster-wise normalization is applied to each sample belonging to the k^{th} cluster to get a set of representations as:

$$\hat{H} = \Bigg[\frac{H[t,l,s,i] - \mu_{k,i}}{\sigma_{k,i} + \varepsilon} \;\Big|\; k = \arg\max_{k} P(z[i] = k | H[t,l,s,i])\Bigg]_{t,l,s,i}$$

where ε is a small constant to preserve numerical stability. Then, we can obtain the Gaussian mixture representation $H^{(\text{gm})} = [H, \hat{H}] \in \mathbb{R}^{T\times L\times S\times 2d_k}$.

Probability Regularization. To enable $P(h[i])$ to describe the empirical data properly, we must regularize the probability of empirical data under $P(h[i])$. Following the convention, we alternatively minimize minus $\log P(h[i])$, equivalent to maximizing the $P(h[i])$.

$$\begin{aligned}\log P(h[i]) =& \log \sum_z P(h[i], z[i]) \\ =& \sum_z P(z[i]|h[i]) \log \frac{P(h[i], z[i])}{P(z[i]|h[i])} \\ =& \mathbb{E}_{P(z[i]|h[i])} \log P(h[i]|z[i]) \\ &- \mathrm{KL}(P(z[i]|h[i]) || P(z[i])) \end{aligned} \tag{6}$$

where $KL(\cdot)$ denotes Kullback–Leibler divergence. However, directly maximizing the LB (in Eq. 6) may cause mode collapsing problem, which means the model cannot distinguish features of each cluster in the expected way [Zhu *et al.*, 2021]. We import the mutual information $I(z[i], h[i])$ into LB [Zhao *et al.*, 2018] to address this issue. Formally, we derive the following cluster objective to minimize:

$$\begin{aligned}\mathcal{L}^{(\text{cluster})} =& \frac{1}{d_k} \sum_i \big[\mathrm{KL}(Q(z[i]) || P(z[i])) \\ &- \mathbb{E}_{P_{\text{data}}(h[i])} \mathbb{E}_{P(z[i]|h[i])} \log P(h[i]|z[i])\big]\end{aligned} \tag{7}$$

where $Q(z[i]) = \mathbb{E}_{P_{\text{data}}(h[i])} P(z[i]|h[i])$ is the expectation of posteriors with respect to the empirical data distribution.

3.3 Temporal Encoder

We implement a Temporal Encoder by applying a set of standard extended 1D convolution filters to extract high-level temporal features. Specifically, our Temporal Encoder consists of three parts: (i) two Dilated Causal Convolutions (DCC) [van den Oord *et al.*, 2016] for achieving a larger receptive field for the representation $H^{(\text{gm})}$, i.e., the receptive field size grows in a linear progression with the depth of the network and the kernel size of the filter; (ii) a Gated Linear Unit (GLU) for introducing the nonlinearity into the network; (iii) a 1×1 convolution function for capturing sequential patterns of time series data. Formally, it works as follows:

$$H^{(\text{te})} = f_{\text{conv}}(\tanh(H^{(\text{gm})} * W_{\text{dc1}}) \odot \sigma(H^{(\text{gm})} * W_{\text{dc2}})) \tag{8}$$

where $*$ denotes the dilated causal convolution, W_{dc1} and W_{dc2} are learnable convolution filters for DCC-1 (filter) and -2 (gate), and $\odot$ denotes an element-wise multiplication operator. $f_{\text{conv}}(\cdot)$ denotes a 1×1 convolution function.

3.4 Hidden Representation Augmenter

Here, we design a *Hidden Representation Augmenter* (HRA) to further enhance the learned representation to be aware of and adaptive to various scenarios. HRA is motivated by a line of research on memory networks ([Wang *et al.*, 2022; Jiang *et al.*, 2022]), which has shown promising results in capturing global patterns. The basic idea of the memory module is to learn a parameterized memory bank that caches global patterns and projects each variable to the same feature space with the memory. The memory module brings two advantages: (i) learning and storing meaningful patterns from a global perspective; (ii) utilizing the knowledge of patterns to construct global representations.

We thereby utilize the idea of a memory module and further put the final representations conditioned on a plugin memory bank $\mathcal{M}$ to encourage the discovery of high-level evolution prototypes, which are representations incorporating the knowledge of the time, the location, and the source variables. Specifically, the memory blank $\mathcal{M} \in \mathbb{R}^{m\times d_m}$ starts by taking the skip connection representation $H^{(\text{sc})} \in \mathbb{R}^{L\times S\times d_k}$ as a query, where m and d_m denote the total number of memory records and dimension of each one. The memory latent representation $H^{(\text{me})}$ can be calculated by:

$$\begin{cases} Q = H^{(\text{sc})} \cdot W_Q + b_Q \\ \phi_j = \dfrac{e^{Q\cdot\mathcal{M}[j]}}{\sum_{j=1}^{m} e^{Q\cdot\mathcal{M}[j]}} \\ V = \displaystyle\sum_{j=1}^{m} \phi_j \cdot \mathcal{M}[j] \\ H^{(\text{me})} = V \cdot W_V + b_V \end{cases} \tag{9}$$

where $Q \in \mathbb{R}^{d_m}$ indicates the query vector projected from flattened $H^{(\text{sc})}$, ϕ_j is the attention score corresponding to j^{th} memory record, and $V \in \mathbb{R}^{d_m}$ is the reconstructed prototype representation. Then, our HRA generates an augmented representation $H^{(\text{aug})} = [H^{(\text{sc})}, H^{(\text{me})}] \in \mathbb{R}^{L\times S\times \overline{(d_k+d_m)}}$ by concatenating $H^{(\text{me})} \in \mathbb{R}^{L\times S\times d_m}$ with $H^{(\text{sc})}$.

3.5 Predictor

The final predictor consists of two fully connected layers with ReLU as an activation function, transforming the dimension of the hidden channel to the desired output dimension:

$$\begin{cases} H^{(\text{out})} = \mathrm{relu}(H^{(\text{aug})}) \cdot W_{\text{o1}} + b_{\text{o1}} \\ \hat{Y} = \mathrm{relu}(H^{(\text{out})}) \cdot W_{\text{o2}} + b_{\text{o2}} \end{cases} \tag{10}$$

where $W_{\text{o1}} \in \mathbb{R}^{(d_k+d_m)\times d_k}$, $W_{\text{o2}} \in \mathbb{R}^{d_k\times O}$, $b_{\text{o1}} \in \mathbb{R}^{d_k}$, and $b_{\text{o2}} \in \mathbb{R}^{O}$ are filter weights and biases. $\mathrm{relu}(\cdot)$ denotes ReLU function that is applied element-wise. $\hat{Y} \in \mathbb{R}^{O\times L\times S}$ indicates the forecasting result of the next O steps for L locations and S sources.

Optimization. In the learning phase, the Mean Squared Error (MSE) is used as the regression loss:

$$\mathcal{L}^{(\text{reg})} = \sum_{t=1}^{O}\sum_{l=1}^{L}\sum_{s=1}^{S} \| Y_{t,l,s} - \hat{Y}_{t,l,s} \|^2 \tag{11}$$

Eq. 7 and Eq. 11 are combined as the loss function for our TTS forecasting model:

$$\mathcal{L} = \mathcal{L}^{(\text{reg})} + \lambda\mathcal{L}^{(\text{cluster})} \tag{12}$$

where $\lambda \geq 0$ is the balancing parameter.

4 Experiments

Dataset	NYC Traffic Demand	BJ Air Quality
Time span	2015/10/24∼2016/1/31	2013/3/1∼2017/2/28
Time slot	1 hour	1 hour
Timestep	4392	35064
Location	98 locations	10 locations
Source	{Bike, Taxi} × {Inflow, Outflow}	PM2.5, PM10, SO_2
# Train/Val/Test	3912/240/240	21048/7008/7008
# Input length	16	16
# Output length	1 /2 /3	1 /2 /3

Table 2: Summary of Experimental Datasets

4.1 Experiment Setup

Datasets. We conduct experiments on two real-world TTS datasets, namely NYC Traffic Demand and BJ Air Quality as listed in Table 2, the details of which are as follows:

- **NYC Traffic Demand dataset**[1 2] is collected from NYC Bike Sharing System, which consists of 98 locations and four sources: Bike Inflow, Bike Outflow, Taxi Inflow, and Taxi Outflow.
- **BJ Air Quality dataset**[3] is collected from the Beijing Municipal Environmental Monitoring Center, which contains 10 locations and three pollutant sources: PM2.5, PM10, and SO_2.

Settings. We implement the network with the Pytorch toolkit. For the model, the number of GMRE-TE layers and cluster components K are set to 4 and 17. The kernel size of each dilated causal convolution component is 2, and the related expansion rate is {2, 4, 8, 16} in each GMRE-TE layer. This enables our model to handle the 16 input steps. The dimension of hidden channels d_z is 24. The parameters for memory bank m and d_m are set to 8 and 48. The batch size is 8, and the learning rate of the Adam optimizer is 0.0001. In addition, the inputs are normalized by Z-Score.

Baselines. To quantitatively evaluate the prediction accuracy of our model, we implement eight baselines for comparison.

- **LSTNet** [Lai *et al.*, 2018]. A deep neural network that combines convolutional neural networks and recurrent neural networks.
- **MTGNN** [Wu *et al.*, 2020b]. A spatial-temporal graph convolutional network that combines graph convolution and dilated convolution.
- **Graph Wavenet** (GWN) [Wu *et al.*, 2019]. A spatial-temporal network that combines adaptive graph convolutions with dilated casual convolution.
- **AGCRN** [Bai *et al.*, 2020]. A recurrent neural network with adaptive graph convolution.
- **StemGNN** [Cao *et al.*, 2020].A spectral temporal graph neural network that maps temp-oral-spatial domain to spectral domain.
- **STtrans** [Wu *et al.*, 2020a]. A deep neural network based on attention mechanism that integrates dynamic spatial, temporal and semantic dependencies.
- **ST-Norm** [Deng *et al.*, 2021]. A normalization-based approach that refines the temporal components and the spatial components.
- **MiST** [Huang *et al.*, 2019]. A co-predictive model for multi-categorical abnormal events forecasting, that models the spatial and categorical embeddings incorporated.

4.2 Overall Performance

We evaluate the performance of our proposed model as well as the above baselines for tensor time series forecasting on all two datasets. We repeat the experiment five times for each model on each dataset and report the average results. Through Table 3, we can find our model outperformed the state-of-the-arts to a large degree. For all four sources of the NYC Traffic Demand dataset, there have improvements of ΔMAE{14.88%, 15.10%, 11.14%, 12.61%}, and ΔRMSE{12.45%, 12.78%, 7.94%, 11.02%} on three horizons average. Among the co-prediction models, MiST performed better than STtrans. However, MiST is second only to MTGNN in terms of bike demand and much inferior in taxi demand. Because it requires reference sources as auxiliary input while outputting the target source prediction, it is not adequate to quantify heterogeneity on three variables (i.e., the time, the location, and the source variable). By contrast, StemGNN and ST-Norm gave a worse performance because ignoring the role of source variables too much makes them not applicable for tensor time series forecasting.

The comparison results with baselines on the BJ Air Quality dataset are shown in Table 4. GMRL has the improvements of ΔMAE{1.87%, 0.90%, 1.45%}, and ΔRMSE{1.05%, 1.66%, 7.94%} for three sources on three horizons average. The BJ Air Quality dataset only has 10 locations and less corrections, and many weather-related features are unavailable. This causes GMRL to perform less well on data with insufficient information, but it fully exploits the correlations among available features. In this case, baseline models are still not as efficient as GMRL.

4.3 Ablation Study

To make a thorough evaluation of key components in our model, we create several variants as follows:

[1] https://ride.citibikenyc.com/system-data
[2] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[3] https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data

Source	Bike Inflow		Bike Outflow		Taxi Inflow		Taxi Outflow	
Method	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTNet	3.21 / 3.32 / 3.66	8.65 / 9.05 / 9.72	3.45 / 3.69 / 3.84	9.40 / 10.14 / 10.43	9.77 / 10.45 / 11.50	20.66 / 21.75 / 23.81	10.44 / 11.11 / 11.87	19.98 / 21.46 / 22.77
AGCRN	2.40 / 2.70 / 3.02	6.96 / 7.69 / 8.41	2.60 / 2.97 / 3.31	7.39 / 8.32 / 9.15	$\underline{7.08}$ / 8.10 / 9.16	13.42 / $\underline{15.71}$ / 18.16	7.81 / 9.03 / 10.19	15.03 / $\underline{17.41}$ / 19.29
MTGNN	$\underline{2.28}$ / $\underline{2.44}$ / $\underline{2.69}$	$\underline{6.68}$ / $\underline{7.05}$ / $\underline{7.62}$	$\underline{2.45}$ / $\underline{2.60}$ / $\underline{2.83}$	$\underline{7.07}$ / $\underline{7.46}$ / $\underline{8.02}$	7.24 / $\underline{8.04}$ / $\underline{8.78}$	13.82 / 15.82 / $\underline{17.71}$	7.97 / $\underline{8.97}$ / $\underline{9.96}$	15.47 / 17.59 / $\underline{19.26}$
GWN	2.37 / 2.66 / 2.91	6.89 / 7.63 / 8.23	2.57 / 2.91 / 3.17	7.42 / 8.26 / 8.85	7.00 / 8.23 / 9.59	$\underline{13.30}$ / 16.11 / 18.99	$\underline{7.72}$ / 9.05 / 10.29	$\underline{14.98}$ / 17.48 / 19.69
StemGNN	2.50 / 2.86 / 3.27	7.17 / 8.04 / 9.01	2.63 / 3.06 / 3.48	7.55 / 8.63 / 9.65	7.80 / 9.38 / 10.84	14.76 / 18.01 / 21.12	8.25 / 9.82 / 11.32	15.83 / 18.89 / 21.62
STtrans	2.72 / 3.06 / 3.38	7.74 / 8.54 / 9.35	2.96 / 3.27 / 3.56	8.33 / 8.93 / 9.67	8.46 / 9.62 / 11.03	15.67 / 17.74 / 20.42	8.78 / 10.04 / 11.28	16.66 / 19.12 / 21.39
ST-Norm	2.37 / 2.64 / 2.91	6.93 / 7.54 / 8.13	2.57 / 2.92 / 3.20	7.42 / 8.29 / 8.95	7.13 / 8.35 / 9.65	13.60 / 16.28 / 18.72	7.88 / 9.17 / 10.44	15.25 / 17.71 / 19.76
MiST	2.32 / 2.52 / 2.76	6.78 / 7.26 / 7.84	2.57 / 2.83 / 3.01	7.48 / 8.10 / 8.54	7.93 / 8.73 / 9.61	14.10 / 16.51 / 19.06	8.44 / 9.19 / 10.82	15.56 / 17.93 / 19.33
GMRL	**2.03 / 2.09 / 2.17**	**6.06 / 6.21 / 6.39**	**2.14 / 2.22 / 2.32**	**6.35 / 6.55 / 6.74**	**6.49 / 7.14 / 7.56**	**12.42 / 14.40 / 16.14**	**7.07 / 7.79 / 8.34**	**13.94 / 15.26 / 16.61**

Table 3: Performance on NYC Traffic Demand Dataset for the first/second/third horizon

Source	PM2.5		PM10		SO_2	
Method	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTNet	12.98 / 17.55 / 21.76	24.09 / 31.56 / 40.11	19.23 / 24.93 / 30.26	31.70 / 39.96 / 52.25	2.15 / 3.69 / 4.47	7.13 / 10.07 / 11.41
AGCRN	9.85 / 15.39 / 19.83	18.74 / 28.15 / 35.06	15.62 / $\underline{22.37}$ / **27.32**	26.41 / 36.13 / $\underline{43.13}$	$\underline{1.57}$ / 2.38 / 3.00	$\underline{6.06}$ / 7.81 / 9.09
MTGNN	9.82 / 14.95 / **19.26**	$\underline{18.41}$ / $\underline{27.35}$ / $\underline{34.10}$	15.83 / 22.70 / 27.61	26.30 / $\underline{36.13}$ / 43.14	1.73 / $\underline{2.31}$ / $\underline{2.82}$	6.39 / $\underline{7.68}$ / $\underline{8.80}$
GWN	$\underline{9.78}$ / $\underline{15.39}$ / 19.72	18.57 / 28.08 / 35.10	$\underline{15.50}$ / 22.39 / 27.43	$\underline{26.09}$ / 36.22 / 43.76	1.61 / 2.45 / 2.95	6.11 / 7.89 / 9.06
StemGNN	10.42 / 16.30 / 21.25	19.53 / 29.30 / 36.71	16.48 / 23.57 / 28.86	27.71 / 37.84 / 45.29	2.47 / 3.13 / 3.51	8.11 / 9.40 / 10.23
STtrans	10.55 / 15.46 / 19.56	21.88 / 31.37 / 38.71	16.61 / 23.14 / 28.21	30.54 / 42.60 / 50.85	2.34 / 3.13 / 3.79	7.55 / 9.41 / 10.71
ST-Norm	13.16 / 17.66 / 22.14	21.42 / 29.98 / 36.89	18.11 / 24.06 / 28.97	28.02 / 37.57 / 44.79	3.47 / 3.42 / 4.32	9.04 / 9.37 / 10.87
MiST	13.98 / 17.77 / 22.07	21.66 / 30.77 / 35.95	18.71 / 24.12 / 29.48	28.20 / 38.44 / 46.42	2.45 / 3.23 / 3.78	7.95 / 9.75 / 10.90
GMRL	**9.53 / 14.80** / $\underline{19.41}$	**18.29 / 26.89 / 33.82**	**15.47 / 21.77** / $\underline{27.37}$	**25.45 / 35.48 / 42.82**	**1.55 / 2.28 / 2.77**	**5.88 / 6.61 / 8.19**

Table 4: Performance on BJ Air Quality Dataset for the first/second/third horizon

Source	Bike Inflow	Bike Outflow	Taxi Inflow	Taxi Outflow
Variant	MAE / RMSE	MAE / RMSE	MAE / RMSE	MAE / RMSE
w/o GMRE	2.65 / 7.25	2.71 / 7.45	9.16 / 18.00	9.40 / 17.87
w/ KMeans	3.23 / 8.84	3.27 / 8.93	10.63 / 20.48	10.82 / 20.42
w/o HRA	2.42 / 6.71	2.55 / 7.06	7.95 / 16.51	8.81 / 16.95
w/o $\mathcal{L}^{(\text{cluster})}$	2.28 / 6.64	2.41 / 7.02	7.93 / 16.37	8.54 / 16.86
GMRL	**2.17 / 6.39**	**2.32 / 6.74**	**7.56 / 16.14**	**8.34 / 16.61**

Table 5: Ablation Study

- **w/o GMRE.** It removes GMRE in each GMRE-TE layer and keeps other components of GMRL, i.e., TTSE is directly used as input to Temporal Encoder.
- **w/ KMeans.** It applies the KMeans clustering algorithm instead of GMRE to obtain a cluster-wise representation, which randomly initializes K cluster centers and calculates the categories according to the distance between the samples and the centers.
- **w/o HRA.** It removes HRA in GMRL, i.e., $H^{(\text{aug})} = H^{(\text{sc})}$, which means the representation $H^{(\text{sc})}$ is directly fed into Predictor to generate the predictions.
- **w/o $\mathcal{L}^{(\text{cluster})}$.** It removes $\mathcal{L}^{(\text{cluster})}$ in GMRE (Eq. 7), i.e., $\mathcal{L} = \mathcal{L}^{(\text{reg})}$ in the learning phase.

Through Table 5, we can see that compared to w/o GMRE, w/ KMeans will deteriorate the forecasting performance, because it only assembles instances by hard clustering, and the clustering result is affected by the selection of the initial cluster center. Removing HRA and $\mathcal{L}^{(\text{cluster})}$ deteriorate performance to a lesser degree. But still, there is a significant gap compared with the full GMRL. All these demonstrate that the proposed GMRL is a complete and indivisible set.

4.4 Case Study

Hidden Representations before/after Cluster Norm

In this subsection, we conduct multiple studies to qualitatively analyze the representations learned from GMRE of the last GMRE-TE layer. In Figure 3, we visualize variations of clusters in different channels before and after applying Cluster Norm. In Figure 3a, the evolution curves of clusters 2, 13, and 14 in Channel 1 overlap and intertwine with each other, especially for clusters 13 and 14. This is because clusters 13 and 14 both originate from the same source (Bike Inflow), and the overlapping features correspond to several identical locations. As shown in Figure 3b, after Cluster Norm, the evolution curves of clusters in Channel 1 become distinguishable. The clusters corresponding to spatial identifiers maintain a stable behavior over time, which remains unchanged even after Cluster Norm. The performance of clusters in Channel 8 before and after Cluster Norm is similar to that in Channel 1, as shown in Figure 3c and Figure 3d. However, in Channel 8, clusters 2, 9, and 16 represent temporal identifiers, and they exhibit changes with varying frequencies over time compared to Channel 1. Clearly, after applying Cluster Norm, each cluster in different channels exhibits heterogeneous evolution with sufficient discriminability. This study confirms the com-

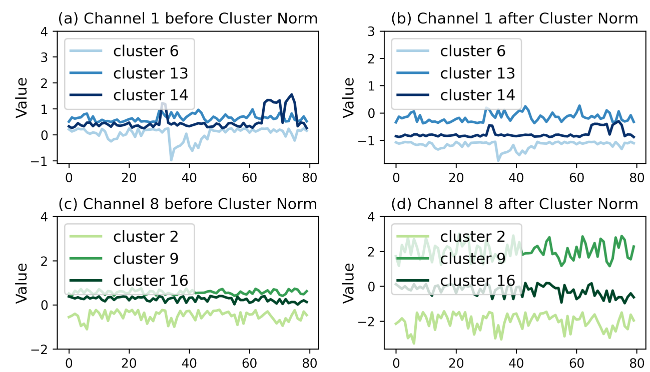

Figure 3: Case studies on cluster variations in different channels before and after Cluster Norm.

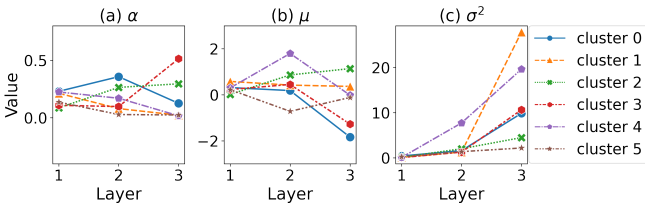

Figure 4: Case studies on the Gaussian mixture parameters α, μ, and σ^2 in each GMRE-TE layer.

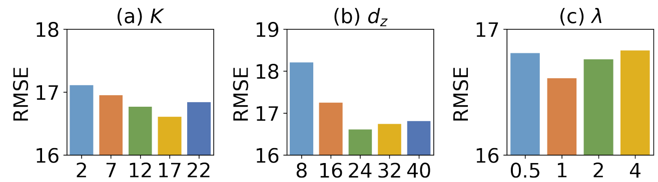

Figure 5: Hyper-parameter studies for the number of cluster components K, the dimension of hidden channels d_z, and the balancing parameter λ.

ponent clustering capability and time-adaptability of GMRL.

Gaussian Mixture Parameters

In Figure 4, we display multiple Gaussian mixture parameters learned in GMRE. As the GMRE-TE layer goes deeper, Gaussian mixture parameters α, μ, and σ^2 can perfectly describe six heterogeneous components after passing through the last GMRE-TE layer. For example, cluster components 4 (purple) and 5 (brown) tend to be similar on parameters α and μ, but their variances are very different. When the models converge, the Gaussian distribution they determine is sufficient to represent heterogeneous components.

4.5 Hyper-parameter Study

We further study the effect of key hyper-parameters manually set in the proposed GMRL, including the number of cluster components K, the dimension of the hidden channel d_z, and the balance parameter λ, as shown in Figure 5. We find that as K and λ increase, the prediction error of GMRL first decreases and then increases. This phenomenon follows our understanding that the model can suffer from underfitting or overfitting due to improper settings of cluster components and balance parameter. Surprisingly, the dimension of the hidden channel has little effect on GMRL: setting it to 24 can achieve fairly good performance, and increasing the dimension of the hidden channel hardly changes the performance.

5 Related Work

Time series forecasting has been studied for decades and applied to various fields. Some studies utilized the attention mechanism for dealing with lost memories [Li *et al.*, 2019; Vaswani *et al.*, 2017; Zhou *et al.*, 2021]. To extract complex spatial-temporal patterns, existing works applied various operations over temporal and spatial domains. Specifically, the graph neural network is developed to model the spatial relation [Yu *et al.*, 2017; Li *et al.*, 2017; Oreshkin *et al.*, 2021; Shang *et al.*, 2021; Chen *et al.*, 2020]; the attention mechanism and its variants contain spatial attention [Fang *et al.*, 2019; Zheng *et al.*, 2020] and temporal attention [Wu *et al.*, 2021; Zheng *et al.*, 2020; Liu *et al.*, 2021; Zhou *et al.*, 2022]; the convolution operator is leveraged to spatial convolution [Deng *et al.*, 2022b; Guo *et al.*, 2021], temporal convolution [Wu *et al.*, 2020b; Wu *et al.*, 2019], spatial-temporal convolution [Guo *et al.*, 2019; Yang *et al.*, 2021] and adaptive convolution [Pan *et al.*, 2019]. [Zhu *et al.*, 2021] proposes an end2end mixture model to cluster microscopic series for macroscopic time series forecasting. [Jiang *et al.*, 2021] reviews the deep learning models and builds a standard benchmark. [Deng *et al.*, 2021; Deng *et al.*, 2022a] refine the high-frequency and local components from MTS data by using normalization technology. Unlike MTS involved data within a matrix, data points within TTS usually take on more substantial heterogeneity and dynamics. Since such unique properties of TTS compared to MTS, traditional methods for predicting MTS (or MTS with reference source) are not entirely suitable for TTS prediction. The inability to explicitly learn and encode interactions from TTS brings up some intuitive concerns. A new line of research further extends the spatial-temporal dependency modeling by considering another domain (i.e., source or categorical domain) [Ye *et al.*, 2019; Huang *et al.*, 2019; Huang *et al.*, 2021; Han *et al.*, 2021; Wang *et al.*, 2021] or extra data source [Jiang *et al.*, 2023]. Our work distinguishes itself from these methods in the following aspects: (i) our GMRE is dynamic, which could explicitly represent the complicated heterogeneous components among multiple variables in TTS; (ii) our HRA enhances the learned representations with sequence-specific global patterns to distinguish and generalize to diverse scenarios, which could help model adaptability to TTS.

6 Conclusion

In this paper, we develop a dynamic Gaussian mixture representations learning framework GMRL, which seeks to individually model each heterogeneity component implied in multiple variables. In addition, we design an HRA module that enhances the learned representation with sequence-specific global patterns to distinguish and generalize to diverse scenarios. Extensive experiments on two real-world datasets verify our perspective's insightfulness and the proposed approach's effectiveness. In the next step, we plan to improve the speed of adaptation and efficiency of GMRL.

Contribution Statement

Renhe Jiang is the corresponding author for this study.

References

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

[Cao *et al.*, 2020] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

[Chen *et al.*, 2020] Weiqi Chen, Ling Chen, Yu Xie, Wei Cao, Yusong Gao, and Xiaojie Feng. Multi-range attentive bicomponent graph convolutional network for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3529–3536, 2020.

[Deng *et al.*, 2021] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. 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*, pages 269–278, 2021.

[Deng *et al.*, 2022a] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Deng *et al.*, 2022b] Leyan Deng, Defu Lian, Zhenya Huang, and Enhong Chen. Graph convolutional adversarial networks for spatiotemporal anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2416–2428, 2022.

[Duan *et al.*, 2022] Jufang Duan, Yi Wang, and Wei Zheng. Plae: Time-series prediction improvement by adaptive decomposition. In *Pacific Rim International Conference on Artificial Intelligence*, pages 394–407. Springer, 2022.

[Fang *et al.*, 2019] Shen Fang, Qi Zhang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Gstnet: Global spatial-temporal network for traffic flow prediction. In *IJCAI*, pages 2286–2293, 2019.

[Fang *et al.*, 2021] Ziquan Fang, Lu Pan, Lu Chen, Yuntao Du, and Yunjun Gao. Mdtp: a multi-source deep traffic prediction framework over spatio-temporal trajectory data. *Proceedings of the VLDB Endowment*, 14(8):1289–1297, 2021.

[Guo *et al.*, 2019] Shengnan Guo, Youfang Lin, Shijie Li, Zhaoming Chen, and Huaiyu Wan. Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3913–3926, 2019.

[Guo *et al.*, 2021] Kan Guo, Yongli Hu, Yanfeng Sun, Sean Qian, Junbin Gao, and Baocai Yin. Hierarchical graph convolution network for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 151–159, 2021.

[Han *et al.*, 2021] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 547–555, 2021.

[Huang *et al.*, 2019] Chao Huang, Chuxu Zhang, Jiashu Zhao, Xian Wu, Dawei Yin, and Nitesh Chawla. Mist: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting. In *The World Wide Web Conference*, pages 717–728, 2019.

[Huang *et al.*, 2021] Chao Huang, Chuxu Zhang, Peng Dai, and Liefeng Bo. Cross-interaction hierarchical attention networks for urban anomaly prediction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4359–4365, 2021.

[Jiang *et al.*, 2021] Renhe Jiang, Du Yin, Zhaonan Wang, Yizhuo Wang, Jiewen Deng, Hangchen Liu, Zekun Cai, Jinliang Deng, Xuan Song, and Ryosuke Shibasaki. Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 4515–4525, 2021.

[Jiang *et al.*, 2022] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. *arXiv preprint arXiv:2211.14701*, 2022.

[Jiang *et al.*, 2023] Renhe Jiang, Zhaonan Wang, Yudong Tao, Chuang Yang, Xuan Song, Ryosuke Shibasaki, Shu-Ching Chen, and Mei-Ling Shyu. Learning social meta-knowledge for nowcasting human mobility in disaster. In *Proceedings of the ACM Web Conference 2023*, pages 2655–2665, 2023.

[Jing *et al.*, 2021] Baoyu Jing, Hanghang Tong, and Yada Zhu. Network of tensor time series. In *Proceedings of the Web Conference 2021*, pages 2425–2437, 2021.

[Lai *et al.*, 2018] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

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

[Li *et al.*, 2019] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[Liu *et al.*, 2021] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

[Oreshkin *et al.*, 2021] Boris N Oreshkin, Arezou Amini, Lucy Coyle, and Mark Coates. Fc-gaga: Fully connected gated graph architecture for spatio-temporal traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9233–9241, 2021.

[Pan *et al.*, 2019] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1720–1730, 2019.

[Rogers *et al.*, 2013] Mark Rogers, Lei Li, and Stuart J Russell. Multilinear dynamical systems for tensor time series. *Advances in Neural Information Processing Systems*, 26, 2013.

[Shang *et al.*, 2021] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021.

[Shao *et al.*, 2022] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting, 2022.

[van den Oord *et al.*, 2016] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, 2016.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2021] Zhaonan Wang, Renhe Jiang, Zekun Cai, Zipei Fan, Xin Liu, Kyoung-Sook Kim, Xuan Song, and Ryosuke Shibasaki. Spatio-temporal-categorical graph neural networks for fine-grained multi-incident co-prediction. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 2060–2069, 2021.

[Wang *et al.*, 2022] Zhaonan Wang, Renhe Jiang, Hao Xue, Flora D Salim, Xuan Song, and Ryosuke Shibasaki. Event-aware multimodal mobility nowcasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4228–4236, 2022.

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

[Wu *et al.*, 2020a] Xian Wu, Chao Huang, Chuxu Zhang, and Nitesh V Chawla. Hierarchically structured transformer networks for fine-grained spatial event forecasting. In *Proceedings of The Web Conference 2020*, pages 2320–2330, 2020.

[Wu *et al.*, 2020b] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[Yang *et al.*, 2021] Song Yang, Jiamou Liu, and Kaiqi Zhao. Space meets time: Local spacetime neural network for traffic flow forecasting. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 817–826. IEEE, 2021.

[Ye *et al.*, 2019] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, Xinran Tong, and Hui Xiong. Co-prediction of multiple transportation demands based on deep spatio-temporal neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 305–313, 2019.

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

[Zhang *et al.*, 2016] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. Dnn-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 1–4, 2016.

[Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[Zhao *et al.*, 2018] Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. *arXiv preprint arXiv:1804.08069*, 2018.

[Zheng *et al.*, 2020] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1234–1241, 2020.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pages 27268–27286. PMLR, 2022.

[Zhu *et al.*, 2021] Zhibo Zhu, Ziqi Liu, Ge Jin, Zhiqiang Zhang, Lei Chen, Jun Zhou, and Jianyong Zhou. Mixseq: Connecting macroscopic time series forecasting with microscopic time series data. *Advances in Neural Information Processing Systems*, 34:12904–12916, 2021.