A Multi-View Multi-Task Learning Framework for Multi-Variate Time Series Forecasting

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang, Fellow, IEEE

Abstract—Multi-variate time series (MTS) data is a ubiquitous class of data abstraction in the real world. Any instance of MTS is generated from a hybrid dynamical system and their specific dynamics are usually unknown. The hybrid nature of such a dynamical system is a result of complex external attributes, such as geographic location and time of day, each of which can be categorized into either spatial attributes or temporal attributes. Therefore, there are two fundamental views which can be used to analyze MTS data, namely the spatial view and the temporal view. Moreover, from each of these two views, we can partition the set of data samples of MTS into disjoint forecasting tasks in accordance with their associated attribute values. Then, samples of the same task will manifest similar forthcoming pattern, which is less sophisticated to be predicted in comparison with the original single-view setting. Considering this insight, we propose a novel multi-view multi-task (MVMT) learning framework for MTS forecasting. Instead of being explicitly presented in most scenarios, MVMT information is deeply concealed in the MTS data, which severely hinders the model from capturing it naturally. To this end, we develop two kinds of basic operations, namely task-wise affine transformation and task-wise normalization, respectively. Applying these two operations with prior knowledge on the spatial and temporal view allows the model to adaptively extract MVMT information while predicting. Extensive experiments on three datasets are conducted to illustrate that canonical architectures can be greatly enhanced by the MVMT learning framework in terms of both effectiveness and efficiency. In addition, we design rich case studies to reveal the properties of representations produced at different phases in the entire prediction procedure.

Index Terms—Time series forecasting, deep learning, normalization, multi-view multi-task learning

1 INTRODUCTION

Time series forecasting is a significant problem in many industrial and business applications [1], [2], [3]. For instance, a public transport operator can allocate sufficient capacity to mitigate the queuing time in a region in advance, if they have the means to foresee that a particular region will suffer from a supply shortage in the next couple of hours [4], [5], [6], [7]. As another example, an investor can avoid economic loss with the assistance of a robo-advisor which is able to predict a potential market crash [8]. Due to the complex and continuous fluctuation of impacting factors, real-world time series tends to be extraordinarily non-stationary, that is, exhibiting diverse dynamics. For instance, traffic volume is largely affected by the road’s condition, location, and the current time and weather condition. In the retail sector, the current season, price and brand are determinants for the sales of merchandise. The diverse dynamics impose an enormous challenge on time series forecasting. In this work, we study multi-variate time series forecasting, where multiple variables evolve with time.

Traditional time series forecasting algorithms, such as ARIMA and state space models (SSMs), provide a principled framework for modeling and learning time series patterns. However, these algorithms have a rigorous requirement for the stationarity of a time series, which suffer from severe limitations in practical use if most of the impacting factors are unavailable. Recent studies show that thanks to the nonlinearity of activation functions, deep learning models possess the capacity to handle complex dynamics, theoretically in any form, even in the absence of additional impacting factors [9]. Therefore, the nonstationarity issue can be addressed to some degree. Common neural architectures applied on time series data include recurrent neural networks (RNNs), long-short term memory (LSTM) [10], Transformer [11], Wavenet [12] and temporal convolution networks (TCNs) [13].

In our article, we conjecture that MTS forecasting can essentially be treated as a multi-view multi-task learning problem. To the best of our knowledge, we are the first to formulate MTS in this way. There are typically two additional views in the MTS problem apart from the original spatial-temporal view, namely the temporal view and the spatial view. From each of these two views, we can divide the forecasting...
for samples into different tasks based on certain criteria. We take shared bike demand as a concrete example, and display the demand data collected from three regions over a five-day period in Fig. 1. In this example, from the temporal view, forecasting at a particular time over all the regions can be grouped into a task; from the spatial view, forecasting at all times over a particular region can be grouped into a task. Herein, the task partition scheme follows the principle where data points sampled from the same task are impacted by a common external factor, which makes them display common patterns. By handling each task with an exclusive predictor, the prediction complexity can be largely reduced in contrast with using a single predictor for all tasks. Sometimes, if the diversity of tasks suitably coincides with the diversity of dynamics, a linear regression model has the sufficient ability to undertake individual tasks.

Distinct from a majority of previous multi-view learning problems whose objectives are to identify the shared information among multiple views [14], [15], our formulation resorts to the supplementary information presented by the additional views. In particular, the temporal view is advantageous at capturing abrupt changes, e.g., due to weather conditions, while the spatial view is beneficial to capturing local patterns which are stable over time. Therefore, these two views can reinforce forecasting from different aspects.

To achieve multi-task learning given any view, the key is to produce a feature space manifesting inter-task weak-correlation and intra-task strong-correlation. To make this idea more comprehensible, we start by displaying the canonical paradigm of multi-task learning in Fig. 2a with its two equivalent derivatives in Figs. 2b and 2c. In Fig. 2b, we create an augmented feature space with the number of dimensions being as many as three times that of the original ones. The augmented feature space is equally partitioned into three subspace where each subspace is associated with a task. Given a sample from any task, we let the corresponding subspace of its belonging task accommodate its features and the other two subspace padded with 0. It is easy to testify the equivalence between Figs. 2a and 2b. Next, let we have a closer look at Fig. 2b. An immediate judgement can be made from this formulation that samples from different tasks are orthogonal to each other in the augmented feature space, and samples from the same task maintain their relative positions as in the original feature space. More generally, even if orthogonality is not rigorously satisfied, each task can be captured in a more individual way provided that the correlations between different tasks diminish. In a further step, we can deduce that given the condition that the inter-task weak correlation and the intra-task strong correlation are manifested in the feature space, the predictor will automatically differentiate the task identity of every given sample. Therefore, all we need is an augmented feature space encoding the two types of relationships as shown in Fig. 2c. In addition, we display two kinds of undesirable geometries in Fig. 3 to highlight the key properties of the geometry which fits the multi-task learning paradigm.

However, only using raw time series data as input features does not obtain the inter-task weak-correlation and the intra-task strong-correlation from either of the spatial view or the temporal view. Although some tasks are inherently separated (i.e., 9am versus 6pm) based on the sequential pattern, most tasks are indistinguishable in the feature space. Following our previous work [16], there are two types of indistinguishability due to the strong inter-task correlation from the spatial view and the temporal view: (1) Spatial indistinguishability means that the dynamics yielded by different variables are not adequately discernible. For instance, looking at the three regions in Fig. 1, we consider their dynamics measured between 8pm and 9pm on different days. In Fig. 4a, we plot the measurement at 8pm versus the measurement at 9pm over the three regions, where the data points are colored in accordance with their regional identities. Different clusters of dynamics are supposed to be distinguishable. However, the cluster-wise relationships (indicated by the direction of a straight line fitting the intra-cluster data points) are highly correlated, which signifies the inter-task strong correlation; (2) Temporal indistinguishability means that dynamics measured at specific times are not substantially discrete. In Fig. 2b, we only plot the
We propose a novel MVMT learning framework for task-wise affine transformation and task-wise normalization, each of which can weaken the inter-task correlation while maintaining the intra-task correlation. Task-wise affine transformation transforms the representations of each sample with task-specific affine parameters, hence task-specific characteristics can be encoded into the feature space. The limitation of this operation is that it can only be applied on the spatial view whose task partition is static over time, or in other words the set of tasks does not change with time. When it comes to the temporal view with dynamic task partition, the model cannot pre-learn the affine parameters for tasks appearing at a future time. To complement task-wise affine transformation, task-wise normalization is proposed, which can be applied not only on the spatial view but also on the temporal view. Basically, it performs normalization over the entire group of samples divided into the same task, which can also result in representations with task-specific characteristics. In our study, we realize task-wise affine transformation and normalization from the spatial view and the temporal view respectively, giving rise to a compound operation known as ST-Norm, abbreviated as STN.

We summarize our contributions as follows:

- We propose a novel MVMT learning framework for time series forecasting. We account for three views in this framework, namely the original view, the spatial view and the temporal view. From each of the spatial view and the temporal view, learning is performed in a multi-task manner where each task is associated with a variable or a timestamp.
- We develop task-wise affine transformation and normalization to enable the feature space to be encoded with MVMT information. Either of these two operations can weaken the inter-task correlations while keeping the intra-task correlations in the representation space, which emulates the explicit partitioning of data samples as in the normal setting of multi-task learning.
- We propose a compound operation ST-Norm, consisting of different realizations of task-wise affine transformation and normalization respectively from the spatial view and the temporal view.
- We conduct extensive experiments to quantitatively and qualitatively validate the effectiveness of the MVMT learning framework.

### 2 RELATED WORK

#### 2.1 Time Series Forecasting

Time series forecasting has been studied for decades. Traditional methods, such as ARIMA, can only learn the linear relationship among different timesteps, which has an inherent deficiency in fitting many real-world time series data that are highly nonlinear. With the power of deep learning models, a large volume of work in this area has recently achieved impressive performance. For instance, [17] adopt LSTM to capture the nonlinear dynamics and long-term dependencies in time series data. However, the memorizing capacity of LSTM is still restricted, as pointed out by [18].

To resolve this issue, [19], [20] create an external memory to explicitly store some representative patterns that can be frequently observed in the history, which is able to effectively guide the forecasting when similar patterns occur. [21] makes use of a skip connection to enable the information to be transmitted from distant history. The attention mechanism is another option to deal with the vanishing memory problem [22], [23]. Of these methods, Transformer is a representative architecture which consists of only attention operations [24]. To overcome the computation bottleneck of canonical Transformer, [11] proposes a novel mechanism that periodically skips some timesteps when performing attention. As far as we know, WaveNet [12], TCN [13] and Transformer [24] are currently the superior choices for modeling long-term time series data [25], [26], [27], [28]. Contrasting to the above approaches focusing on enriching the semantics of the current state with the information extracted from the history, [29] attempted to improve the emissions from the current state into the future. They formulated the forecasts at different forthcoming time steps as separate tasks and modeled the interplay between them.

To tackle MTS, several studies [27], [30], [31], [32] assume that multi-variate time series data has a low-rank structure. Another thread of works [17], [33], [34] leverages the attention mechanism to learn the correlations among individual time series, where [34] opted to obtain the attentive scores with cosine similarity other than [17], [33] using dot product similarity. Recently, [26] inferred the inherent structure over the variables derived from self-learned encodings associated with each variable. [35] used Fourier transform to decompose original MTS data into a group of orthogonal signals. [36] proposed a dual self-attention network (DANet) to dynamically capture both local and global patterns with convolution and attention operators, effectively handling MTS data with non-periodic dynamics. The above methods make point estimation. [30], [37], [38] propose a confidence interval that is likely to contain the forthcoming emissions from the current state into the future. They formulated the forecasts at different forthcoming time steps as separate tasks and modeled the interplay between them.

Contrasting to the above approaches focusing on enriching the semantics of the current state with the information extracted from the history, [29] attempted to improve the emissions from the current state into the future. They formulated the forecasts at different forthcoming time steps as separate tasks and modeled the interplay between them.
connectiveness is characterized as the quantity of the traffic flow between the OD pair [39]. [40] proposed a new objective function for training deep neural networks, aiming at accurately predicting sudden changes. [41] explored the structure of LSTM to learn variable-wise hidden states, with the aim to distinguish the contribution of variables to the prediction. In some scenarios, MTS may evolve asynchronously and are spaced unevenly. To deal with this more general case, [42] organized the asynchronous MTS data as a single series of observations, alongside a series of external features expressing the spatial-temporal relationships between the observations. Afterward, they married convolutional neural networks and auto-regressive models to perform forecasting on top of this new data representation.

Distinguishing from the existing works, we are the pioneer in developing spatial and temporal normalization in the context of MTS forecasting, and demonstrate their rationality from the perspective of multi-view multi-task learning.

### 2.2 Normalization

Normalisation was first adopted in deep image processing, and has significantly enhanced the performance of deep learning models for nearly all tasks. There are multiple normalisation methods, such as batch normalisation [43], instance normalisation [44], group normalisation [45], layer normalisation [46] and positional normalisation [47], each of which is proposed to address a particular group of computer vision tasks. Of these, instance normalisation has the greatest potential for our study, which was originally designed for image synthesis owing to its power to remove style information from the images. Researchers have found that feature statistics can capture the style of an image, and the remaining features upon normalizing the statistics are responsible for the content. Such a separable property enables the content of an image to be rendered in the style of another image, which is also known as style transfer. The style information in the image is like the scale information in time series. There is another line of work which explores the reason why the normalization trick facilitates the learning of deep neural networks [48], [49], [50], [51].

### 3 PRELIMINARIES

In this section, we introduce the definitions and the assumptions. All frequently used notations are reported in Table 1.

#### 3.1 Task Partition Schemes

Given the complete set of sample indexes \( \mathcal{I} = [1, N] \times [1, T] \), a task is defined as \( \mathcal{P} = \{I_1, \cdots, I_M\} \), where \( I_1, \cdots, I_M \) are disjoint subsets (also known as tasks in our context) of \( \mathcal{I} \), and \( M \) is the number of tasks been partitioned. Besides, we introduce a function \( \mathcal{P} \) which maps any sample index to the index of its belonging task \( \mathcal{P}(n, t) = m \) under the task partition scheme.

We employ two schemes to partition tasks respectively from the spatial view and the temporal view, which are introduced as follows:

- **Temporal view:** samples collected at the same time are put in the same task as shown in Fig. 5a, ignoring the spatial difference. In this case, \( M = T \) and \( I_m = \{(n, m)\}_{n=1}^T \).
- **Spatial view:** samples of the same variable are put in the same task as shown in Fig. 5b, ignoring the temporal difference. In this case, \( M = N \) and \( I_m = \{(m, t)\}_{t=1}^T \).

In terms of any of the aforementioned partition schemes, samples put in the same task are impacted by the same factor, thus they should manifest a strong correlation. Meanwhile, samples across tasks are impacted by different factors, which result in weakly correlated patterns.

There are many partition schemes with different granularities. For example, we can also put the samples collected during a specified period in a task. In our work, we adopt the finest granularity, and the question of what the optimal granularity is is left to explore in future works.

#### 3.2 Other Preliminaries

**Definition 1 (Time series forecasting).** Time series forecasting is formulated as the following conditional distribution:

\[
P(X_{t+1:T_{out}} | X_{t+1:T_{in}}) = \prod_{t=1}^{T_{out}} P(X_{t+1} | X_{t+1:T_{in}}),
\]

**Definition 2 (Time series factorization).** Specifying a task partition scheme \( \mathcal{P} \), any sample of the time series can be factorized in the following way:
\[ Z_{n,t} = G_m^P L_{n,t}^P, \]  

where \( m = \mathcal{P}(n, t) \) denotes the task belonging to the sample under \( \mathcal{P} \), \( G_m^P \) is a global component shared by all the samples from task \( m \) under \( \mathcal{P} \), and \( L_{n,t}^P \) is a local component only possessed by sample \( (n, t) \). Be aware that different task partition schemes will result in different forms of factorization.

**Assumption 1.** We postulate that different dimensions of the local component are independent, and they follow a multi-variate normal distribution as follows:

\[
L_{n,t}^P \sim \mathcal{N} \left( \begin{bmatrix} \beta_m^P[0] \\ \vdots \\ \beta_m^P[d-1] \end{bmatrix}, \begin{bmatrix} (y_m^P[0])^2 & \cdots & (y_m^P[d-1])^2 \\ \vdots & \ddots & \vdots \\ (y_m^P[0])^2 & \cdots & (y_m^P[d-1])^2 \end{bmatrix} \right),
\]

where \( \beta_m^P \) denotes the mean vector and \( \beta_m^P[i] \) denotes its \( i \)th entry; the co-variance matrix is a diagonal matrix, where the vector of entries on the main diagonal are denoted as \( (y_m^P)^2 \) and all the off-diagonal entries are 0.

**Remark 1.** We assume the off-diagonal entries of the covariance matrix to be 0, in the sense that this setup simplifies the form of time series factorization and the following analysis. Such kind of simplification facilitates revealing the rationale of our method. For a more complicated setup where different dimensions are correlated with each other, some sophisticated whitening operations can be applied into use [52], [53], which is left to be investigated in future work.

**Remark 2.** Given a collection of samples from any task \( m \), their local components are denoted as \( \{L_{n,t}^P|\mathcal{P}(n, t) = m\} \). Intuitively, \( \{L_{n,t}^P|\mathcal{P}(n, t) = m\} \) is expected to span a sub-space whose dimension is lower than the effective dimension of the latent representation space \( d \), as \( L_{n,t}^P \) results from the lower number of environmental factors compared to \( Z_{n,t} \), Hence, only a part of the entries in \( y_m^P \) have non-zero values. In addition, if \( \{L_{n,t}^P|\mathcal{P}(n, t) = m_1\} \) and \( \{L_{n,t}^P|\mathcal{P}(n, t) = m_2\} \) with different task identities are impacted by the same group of environmental factors, they are supposed to span the same subspace.

**Remark 3.** Indistinguishability is attributed to the phenomenon where latent representations from different tasks span the same subspace. In the rest of this paragraph, we discuss the root cause of this phenomenon. To begin with, we can treat Eq. (1) from the view of geometric transformation. In this way, local components from the same task will experience the same transformation relying on the corresponding global component, and the ones from the different tasks will experience different transformations. In practice, there are multiple types of geometric transformations, which are made up of three basic transformations, namely scaling, translating and rotating. Scaling is the one that will not change the space spanned by the local components, so any group of transformations that only differ in scaling factors will cause the produced latent representations to be indistinguishable. A concrete instance of scaling in the real-world is the effect imposed by population: taxi demand over a region is normally proportional to the population residing in this region.

### 4 Methodology

An overview of the multi-view multi-task learning block is displayed in Fig. 6. Any representation input to this block is firstly replicated to three copies. Then, the three copies are separately transformed by specific operations respectively from the spatial view, the temporal view and the original view. Finally, the resulting three copies are concatenated together to obtain an augmented representation, which is taken to be the output of this block.

In this section, we start by introducing the proposed task-wise affine transformation and task-wise normalization respectively in Sections 4.1 and 4.2; then, in Section 4.3, we demonstrate how to alter Wavenet under the guidance of MVMT framework; we conclude this section by introducing the process of forecasting and learning in Section 4.4.

#### 4.1 Task-Wise Affine Transformation

Task-wise affine transformation differentiates tasks by assigning each task with an exclusive group of affine parameters. As each group of parameters is only responsible for capturing the dynamics of a single task, the indistinguishability issue can be mitigated.

Formally, with respect to a task partition scheme \( \mathcal{P} \) and its associated mapping function \( \mathcal{P} \), task-wise affine transformation takes the following operations:

\[
\tilde{Z}_{n,t}^P = Z_{n,t}^P w_m^P + b_m^P,
\]

where \( m \) denotes \( \mathcal{P}(n, t) \); \( w_m^P \) and \( b_m^P \) are two affine parameters targeting at task \( m \) under the partition \( \mathcal{P} \). To be noted that affine transformation is a special case of a fully connected layer, where interactions between the feature channels are taken into consideration. We will empirically show that task-wise affine transformation in conjunction with the following task-wise normalization achieves competitive performance, and introduces far fewer parameters in contrast with the task-wise fully connected layer.

Although task-wise affine transformation allows for more freedom to learn task-wise dynamics, it has severe limitation in handling cold-start tasks. In the practice of time series forecasting, tasks partitioned from the temporal view accumulate with time. For each new task encountered in the testing phase, we must identify a task that not only presents similar dynamics to the one being tested, but it has also been observed in the training data. For a time series
showing regular patterns, such identification can be accomplished given prior knowledge on regularity. Nonetheless, for the ones with irregular patterns, it would be cumbersome to identify eligible tasks.

Due to the limitation, we apply task-wise affine transformation from the spatial view. The implementation takes the following form:

$$Z_{n,t}^S = Z_{n,t}w_n^S + b_n^S,$$

where $w_n^S$ and $b_n^S$ are learnable affine parameters.

Next, we introduce another thread of approaches which can address the cold-start problem.

### 4.2 Task-Wise Normalization

Task-wise normalization explicitly encodes global components into the representation space. The global component varies from task to task, and thus can be used to separate tasks. However, the observation of any sample is a mixture of the global component and the local component, which hinders the capture of the global component. To extract the global component, we start by applying task-wise normalization to eliminate the global component from the representation, then we combine the normalized representation with the original representation to obtain an augmented representation for each sample. The augmented representation space can manifest the difference on global components, and hence the current inter-task correlation is weaker than the original.

Likewise, for a task partition scheme $P$ and its associated mapping function $P$, task-wise normalization is performed as follows:

$$\hat{Z}_{n,t}^P = \frac{Z_{n,t} - \mu_m^P}{\sigma_m^P}$$

where we let $m$ denote $P(n, t)$ which is the task belonging of the sample,

$$\mu_m^P = E[Z_{i,j} \mid P(i,j) = m]$$

$$\sigma_m^P = \sqrt{E[(Z_{i,j} - \mu_m^P)^2 \mid P(i,j) = m]}$$

$$\approx \frac{1}{N} \sum_{i,j} Z_{i,j},$$

$$\approx \frac{1}{N} \sum_{i,j} (Z_{i,j} - \mu_m^P)^2 + \epsilon,$$

where $\epsilon$ is a small constant to preserve numerical stability. By implementing $P$ with different task partition schemes, we gain representations normalized in different ways. For instance, the task partition from the temporal view produces the following implementation of task-wise normalization:

$$Z_{n,t}^T = \frac{Z_{n,t} - \mu_t^T}{\sigma_t^T},$$

where

$$\mu_t^T = \frac{1}{N} \sum_{i=1}^N Z_{i,t},$$

$$\sigma_t^T = \sqrt{\frac{1}{N} \sum_{i=1}^N (Z_{i,t} - \mu_t^T)^2 + \epsilon}.$$

For conciseness, we do not include the implementation from the spatial view.

Next, we explain why task-wise normalization works. As we indicate in Remark 3, certain cases of indistinguishability are caused by task-wise transformations that differ only in scaling factors. Task-wise normalization resolves this issue by converting scaling transformation to rotating transformation. Basically, we construct rotation with a rotating angle which relies on the scaling factor. First, we rewrite the normalized representation from Eq. (4) in the following way:

$$\hat{Z}_{n,t}^P = \frac{Z_{n,t} - \mu_m^P}{\sigma_m^P} = \frac{1}{\gamma_m^P} \frac{Z_{n,t} - \beta_m^P}{\gamma_m^P},$$

where Eq. (8) is deduced by substituting Eq. (1), Eq. (2), Eq. (5) and Eq. (6) into Eq. (7). Then combining the original view, the spatial view and the temporal view, we map the original representation to an augmented representation as follows:

$$Z_{n,t} \rightarrow \frac{Z_{n,t}}{\gamma_m^P} \frac{Z_{n,t} \gamma_m^P - \beta_m^P}{\gamma_m^P},$$

where $m_1 = S(n,t)$ and $m_2 = T(n,t)$. At this step, from the view of geometric transformation, each sample point is rotated by an angle depending on $G_m^P, Y_m^P$ from its original position and is then translated by $\beta_m^P, Y_m^P$. Thus, the difference on $G_m^P, \beta_m^P$ and on $Y_m^P$ can be manifested in the new space. It is noteworthy that the new space also maintains the correlation among $G_m^P$ belonging to different tasks, which can facilitate the learning process.

### 4.3 Wavenet

We illustrate the architecture of our work in Fig. 7. Some key variables with their shapes are labeled at their corresponding positions along the computation path. Generally, our framework is instantiated as a structure like Wavenet [12], except that we incorporate a ST-Norm block into the residual block.

We briefly introduce a dilated causal convolution where the filter is applied with skipping values. For a 1-D signal $z \in \mathbb{R}^T$ and a filter $f : \{0, \ldots, k - 1 \} \rightarrow \mathbb{R}$, the causal convolution on element $i$ is defined as follows:

$$F(t) = (z * f)(t) = \sum_{i=0}^{k-1} f(i) \cdot z_{t-i}.$$
long history. Pooling is a natural choice to address this issue, but it sacrifices the order information presented in the signal. To this end, dilated causal convolution, as shown in Fig. 8, is used, a form which supports the exponential expansion of the receptive field. The formal computing process is written as:

$$F(t) = (z *_d f)(t) = \sum_{i=0}^{k-1} f(i) \cdot z_{i-d+1},$$  

(10)

where $d$ is the dilation factor. Normally, $d$ increases exponentially w.r.t. the depth of the network (i.e., $2^l$ at level $l$ of the network). If $d = 1$ ($2^0$), then the dilated convolution operator $*_{d}$ reduces to a regular convolution operator $*$.

### 4.4 Forecasting and Learning

We let $z^{(L)} \in \mathbb{R}^{N \times T_{in} \times d_{l}}$ denote the output from the last residual block, where each row $z^{(L)} \in \mathbb{R}^{T_{in} \times d_{l}}$ represents a variable. Then, we employ a temporal pooling block to perform temporal aggregation for each variable. Several types of pooling operations can be applied, such as max pooling and mean pooling, depending on the problem being studied. In our case, we select the vector in the most recent time slot as the pooling result, which is treated as the representation of the entire signal. Finally, we make a separate prediction for each variable, based on the obtained representation using a shared fully connected layer.

In the learning phase, our objective is to minimize the mean squared error between the predicted values and ground truth values. We use the Adam optimizer [54] to optimize this target.

The present SOTA is monopolized by models equipped with a graph learning module, which establishes mutual relationships between different time series [25], [26], [55]. As a result, the computational complexity of this module is $O(TN^3)$. In contrast, the task-wise affine transformation and the task-wise normalization modules only involve $O(TN)$ operations, presenting better scalability than graph-based models when the number of nodes is massive.

### 5 Evaluation

In this section, we describe the extensive experiments on three common datasets to validate the effectiveness of MVMT framework from different aspects.

#### 5.1 Experimental Setting

##### 5.1.1 Datasets

We validate our model on five real-world datasets, namely BikeNYC, PeMSD7, Electricity, PM2.5 and Solar energy. The statistics regarding each dataset as well as the corresponding settings of the designed task are reported in Table 2. We standardize the values in each dataset to facilitate training and transform them back to the original scale in the testing phase.

Table 2 reports the statistics of the datasets. More details regarding the datasets are as follows.

- **PeMSD7**\(^1\). The data is collected from the Caltrans Performance Measurement System (PeMS) using sensor stations, which are deployed to monitor traffic speed across the major metropolitan areas of the California state highway system. We further aggregate the data to 30-minute intervals by average pooling.
- **Electricity**\(^2\). The original dataset contains the electricity consumption of 370 points/clients, from which 34 outlier points that contain extreme values are removed. Moreover, we calculate the hourly average consumption for each point, and take it as the time series being modeled.
- **BikeNYC**\(^3\). Each time series in this dataset denotes the aggregate demand for shared bikes over a region in New York City. We do not consider the spatial relationship presented in the PeMSD7 and BikeNYC data, since our objective is to study the temporal patterns.
- **Solar energy**\(^4\). It contains the solar power production records in the year of 2006, which is sampled every 10 minutes from 137 PV plants in Alabama State.
- **PM2.5**\(^5\). It contains hourly PM2.5 data from multiple air-quality monitoring sites, acquired from the Beijing Municipal Environmental Monitoring Center.

1. https://pems.dot.ca.gov/
2. https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams2012014
3. https://ride.citibikenyc.com/system-data
4. http://www.nrel.gov/grid/solar-power-data.html
5. https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data
The time period spans from March 1st, 2013 to February 28th, 2017.

5.1.2 Network Setting
The batch size is 8, and the input length of the batch sample is 16. For the Wavenet backbone, the layer number is set to 4, the kernel size of each DCC component is 2, and the associated dilation rate is $2^i$, where $i$ is the index of the layer (counting from 0). Such settings collectively enable the output from Wavenet to perceive 16 input steps. The number of hidden channels $d_z$ in each DCC is 16. We apply zero-padding on the left tail of the input to enable the length of the output from DCC to be 16 as well. The learning rate of the Adam optimizer is 0.0001\(^6\).

5.1.3 Evaluation Metrics
We validate our model using root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE).

We conduct cross validation on each dataset to get a comprehensive evaluation of our model against competing ones. In particular, we split time series data into 10 chunks along the temporal axis with approximately equal length, then we create 5 groups of training/validation/test sets constructed from these 10 data chunks. For the first group, the first 4 chunks of data construct the training set, while the 5th and the 6th chunk respectively construct the validation and testing set; For the second group, the first 5 chunks construct the training set, while the 6th and the 7th chunk respectively construct the validation and testing set, etc. We repeat the experiment 10 times for each model on each group of training/validation/test set and report the average performance.

5.2 Baseline Models
- **MTGNN** [26]. MTGNN constructs inter-variate relationships by introducing a graph-learning module. Specifically, the graph learning module connects each hub node with its top k nearest neighbors in a defined metric space. MTGNN’s backbone architecture for temporal modeling is Wavenet.
- **Graph Wavenet** [25]. The architecture of Graph Wavenet is similar to MTGNN. The major difference is that the former derives a soft graph where each pair of nodes has a continuous probability of being connected. We also test the performance of TCN and Transformer incorporating STN, where STN is similarly applied before the causal convolution operation in each layer.

5.3 Experimental Results
The experimental results on the five datasets are reported in Table 3. The improvements achieved by Wavenet + STN over the best benchmarks are recorded in the last row of each sub-table.

It is obvious that Wavenet + STN achieves SOTA results over almost all horizons on the BikeNYC, PeMSD7 and electricity data. The reason for this is that we refine the high-frequency components from both the temporal view and the spatial view, which are generally overlooked by baseline models. Next, we reveal the cause of Wavenet + STN’s under-performance on the electricity dataset over the first horizon with respect to MAPE. Electricity data follows a long-tailed distribution – a certain proportion exceeds a relatively high level. Recall that the optimization involves minimizing mean squared error, which means that more weights are placed on large errors. Moreover, every sample is treated equivalently in the estimation of global statistics. Therefore, the model can fit long-tailed samples better, but at the cost of degrading the fitness on normal samples.

We further investigate the improvement in terms of an individual variable in MTS. To prove that STN captures the

**TABLE 2** Dataset Statistics

| Tasks       | Electricity | PeMSD7 | BikeNYC | Solar | PM2.5 |
|-------------|-------------|--------|---------|-------|-------|
| Start time  | 10/1/2014   | 5/1/2012 | 4/1/2014 | 1/1/2006 | 3/1/2013 |
| End time    | 12/31/2014  | 6/30/2012 | 9/30/2014 | 12/31/2006 | 2/28/2017 |
| Sample rate | 1 hour      | 30 minutes | 1 hour | 10 minutes | 1 hour |
| # Variate   | 336         | 228     | 128     | 137    | 10    |
| Input length| 16          | 16      | 16      | 16     | 16    |
| Output length| 3          | 3       | 3       | 3      | 3     |

6. Code available at: https://github.com/JLDeng/ST-Norm.git

...
difference in scaling factors among various variables, we characterize a variable with the mean of its historical observations. For succinctness, we calculate the variable-wise reduction on RMSE obtained by Wavenet + STN compared to AGCRN, and then plot the reduction against the scale of each variable in Fig. 9. We can see that for BikeNYC and electricity, the improvement becomes more prominent as the scale grows, which meets our previous expectation. For PeMSD7, the improvement is less significant, as all the variables in this dataset vary in the same range, which signifies that they have approximately the same scaling factor.

Figs. 10 and 11 illustrate the efficiency of our model. Fig. 11 shows that with the additional STN module, the converging speeds of the models are accelerated by a large margin, faster than nearly all the baseline models. Fig. 10 indicates that the running time of training our model on the same volume of data is also competitive with baselines.

![Table 3](image-url)
5.4 Ablation Study

We design several variants within the MVMT framework as shown in Fig. 12 to validate the effectiveness of different operations. We evaluate these variants on the first three datasets and report the results in Table 4. From this table, we can draw the following major conclusions:

- Taking either the spatial view or the temporal view can greatly enhance the capability of vanilla Wavenet, and taking both achieves the best performance.
- Contrasting (a) with (b), we conclude that the original view is indispensable in the MVMT framework, especially when applied on the electricity data. An intuitive explanation for this indispensability is that task-wise normalization more or less loses information encoded in the original representation, especially for data which presents complex dynamics like the electricity dataset.
- Based on (a), (c) and (e), we find that task-wise affine transformation significantly increases the improvement contributed by the spatial branch. The reason for this is that this operation links the tasks associated with the same region over time, which is paramount for online forecasting where only recent observations are input to the model.
- Based on (a), (d), (e) and (f), we find that the performance gain entailed by merely taking the temporal view is limited on the electricity data. We conjecture that this is also due to the complex patterns of the data: without encoding the spatial attribute into the representations, it is difficult for the model to capture typical temporal patterns from data.

5.5 Hyper-Parameter Analysis

We further study the effect of different settings of the hyper-parameters in the proposed modules. There are four hyper-parameters need to be manually set by practitioners, namely the dimension of hidden channels $d_z$, the number of historical steps input to the model, the kernel size of DCC and the batch size. When evaluating each hyper-parameter above, the remaining three are fixed to their default setting as introduced in Section 5.1.2. The study results are reported in Fig. 13, from which we can draw a major conclusion: STN not only boosts the performance, but also increases the stability of the performance under different hyper-parameter settings.

5.6 Case Study

To obtain more insights on the algorithm, we conduct multiple studies to qualitatively analyze the representations generated while forecasting, including the initial representation, the intermediate representation and the final representation. The dataset we select for this investigation is BikeNYC.

5.6.1 Initial Representation

We apply task-wise normalization over the raw input data respectively from the spatial view and the temporal view, and examine whether the issues we raise in Fig. 2 are mitigated. We plot the original quantity versus the temporally normalized quantity in Fig. 14, and the original quantity versus the spatially normalized quantity in Fig. 15. It is apparent that the pairwise relationship between the original quantity and the temporally normalized quantity separates different regions, and the pairwise relationship between the original quantity and the spatially normalized quantity separates different days.

5.6.2 Intermediate Representation

In this part, we investigate the output from the spatial branch and the temporal branch in STN. We start by discussing the property that these intermediate representations are supposed to express. The output from the spatial branch is expected to encode the local component which reflects the temporary pattern at a single timestamp. The reason for this is that task-wise normalization eliminates the global component from the input representation, where the global component encodes the long-term pattern regarding a region, which has nothing to do with a specific timestamp. With respect to the temporal branch, the resulting
representation is deemed to show the region-wise pattern, which plays the role of a local component from this view.

We select three representative regions at specified times to reflect what the operations extract from the data. We examine the intermediate representation output from the two views in the top residual block. As a comparison, we also inspect their associated input representations, each of which is a concatenation of raw measurements. Next, we discuss separately the outcomes from these two views in details.

Temporal View. In Fig. 16, we display the demand evolution during a given period over the three investigated regions. We can observe that the three regions have similar evolution patterns, especially regions B and C. The representations concatenated by the original measurements are plotted in Fig. 17a, and the output of the intermediate representations from the temporal view are plotted in Fig. 17b. For the sake of visualization, we obtain the two-dimensional embeddings of these representations via t-Distributed Stochastic Neighbor Embedding (t-SNE). We can observe that the representations are completely rearranged in accordance with the regional identity. This observation demonstrates that the local components are roughly invariant within the group belonging to the same region. This coincides with our understanding that some regional attributes, such as population and functionality, are stable over time.

Spatial View. To reflect the characteristics of the output of representations from the spatial view, we take another region D into consideration, as shown in Fig. 18. Noticeably, the magnitude of the demand over region D is substantially

---

**Table 4**

| Region | RMSE | MAE | MAPE (%) |
|--------|------|-----|----------|
| B      |      |     |          |
| RMSE   | 5.32 | 5.40| 6.34     |
| MAE    | 2.60 | 2.65| 3.04     |
| MAPE (%)| 19.1 | 19.5| 22.9     |
| P      |      |     |          |
| RMSE   | 5.25 | 5.38| 5.99     |
| MAE    | 2.88 | 3.01| 3.39     |
| MAPE (%)| 19.1 | 19.5| 22.9     |
| E      |      |     |          |
| RMSE   | 38.9 | 41.0| 43.1     |
| MAE    | 18.8 | 21.1| 21.4     |
| MAPE (%)| 14.2 | 17.7| 16.2     |

---

**Fig. 12. Variants for ablation study.**

---

**Fig. 13. Hyper-parameter analysis.**
smaller than those over regions A or B. Here, we account for three different times in a day, consisting of 1am, 8am and 12pm. Likewise, the input representations are plotted in Fig. 19a, and the intermediate representations are plotted in Fig. 19b. As shown in Fig. 19a, instances belonging to region D are mixed up without separation between different times, which signifies that the model will struggle to differentiate the times at which those instances occurred. In contrast, this issue is mitigated in the representation space from the spatial view, as it forms clusters of instances with the same occurrence time.

5.6.3 Final Representation
In this part, we investigate the representations directly used for prediction. The major purpose is to examine the spatial distinguishability and temporal distinguishability of the representations after processing by STN. These two types of distinguishability are separately illustrated with two cases.

**Temporal View.** In the first case, we examine the distinguishability from the temporal view, concentrating on a specific region. With prior knowledge that bike demand shows daily periodicity where the same evolving pattern appears iteratively at the same time every day, we select four representative times of a day which exhibit entirely different patterns. Then, we extract the input sequences at these times, and visualize them in Fig. 20, grouped by time.
To demonstrate that the distinguishability of representations can be greatly enhanced by the additional operations, we employ raw sequence and representation produced by vanilla Wavenet as two benchmark representations. We firstly perform principal component analysis (PCA) aiming at reducing the dimensionality of the representations. The reason for adopting PCA rather than t-SNE is that PCA can preserve the linearity on the representation space. The percentage of variance explained by each of the selected components is plotted in Fig. 21, from which we can observe that only a few components are effective as the variance

Fig. 21. The percentages of variance explained by the first seven components.

Fig. 22. Projections of representations on the subspace spanned by (a), (c), (e): the first two components; (b), (d), (f): the third component and fourth component.

Fig. 23. Two groups of samples selected to inspect spatial distinguishability.

Fig. 24. The percentage of variance explained by the first seven components.

Fig. 25. Projections of representations on the subspace spanned by the first two components, colored in accordance with (a), (c), (e): group identity and (b), (d), (f): scale.
over the others is extremely small. According to this finding, we visualize in Fig. 22 the projections on the first four components which explain almost all the variance over the representations. It is obvious that the enhanced model produces representations showing the strongest intra-task correlation and the weakest inter-task correlation of the three models. Hence, the distinguishability from temporal view is improved by applying the proposed operations.

**Spatial View.** In the second case, we examine the distinguishability from the spatial view, concentrating on a particular time. We roughly divide the regions into two groups based on the evolving patterns as shown in Fig. 23. Basically, bike demand over the first group of regions experiences a v-shape variation during the period being visualized and will continue rising in the forthcoming time steps; on the contrary, the second group of regions approximately remains constant, e.g., 0. We clarify that the assignment is not rigorous and unique, but such fuzziness will not affect the conclusion we can draw.

Similar to the first case, we perform PCA, plot the percentage of variance explained by each of the selected components in Fig. 24 and visualize in Fig. 25 the projections on the first two components which explain most of the variance. At this step, it is apparent that the enhanced model yields the most distinguishable representation space from the spatial view. Moreover, we recolor the sample points based on the scale of the observations, and the visualization results show that the representations also encode the information regarding scale.

### 6 Conclusion and Future Work

In this work, we develop a novel multi-view multi-task learning framework for multi-variate time series forecasting. In our design, this framework consists of the original view, the spatial view and the temporal view, but it is flexible to account for more views depending on the specific application. Forecasting from each view is accomplished in a multi-task manner via two task-dependent operators, namely task-wise normalization and task-wise affine transformation. Diverse experiments were performed to quantitatively show the effectiveness, the efficiency and the robustness of the framework. Furthermore, we conducted multiple case studies on representations produced at different stages in the forecasting procedure. The outcome qualitatively demonstrates that this framework strengthens the intra-task correlation, while weakening the inter-task correlation.

It is noticeable that the framework is modular, flexible and extensible. In practice, we start with defining a variety of task partition schemes from different views, beyond the spatial and the temporal view, based on the structure underlying the specified data. Then, we instantiate a collection of task-wise affine transformation and normalization modules to accommodate these schemes. For example, if the set of variables is only partially correlated, we can identify the correlated subset and define a scheme to distinguish this subset. Like crafting advantageous features, formulating views and tasks beneficial for forecast relies on prior knowledge, i.e., the practitioners’ insight into the data.

When dealing with variable-length sequences, the prevalent forecasting methods pad or truncate them to the same length at the pre-processing step, which unavoidably introduces noise or abandons valuable samples. In contrast, the MVMT framework can directly process variable-length sequences, as the mean and the standard deviation computations make no difference for different sample sizes.

There are two limitations of this framework. First, whether trialing different schemes can be automated to save efforts in model architecture design is an open problem. Second, it is worth exploring how to apply this framework on streaming data, where data distribution is shifting over time. In this new scenario, the current implementation of the MVMT framework will not be suitable or at least have considerable room to be improved, since the parameters in spatial-view affine transformation cannot adapt to the unknown data distribution.

### References

[1] R. Jiang et al., “DeepCrowd: A deep model for large-scale city-wide crowd density and flow prediction,” *IEEE Trans. Knowl. Data Eng.*, early access, May 03, 2021, doi: 10.1109/TKDE.2021.3077056.

[2] J. Deng, X. Chen, Z. Fan, R. Jiang, X. Song, and I. W. Tsang, “The pulse of urban transport: Exploring the co-evolving pattern for spatio-temporal forecasting,” *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 6, pp. 1–25, 2021.

[3] R. Jiang et al., “DeepUrbanEvent: A system for predicting city-wide crowd dynamics at big events,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2114–2122.

[4] X. Geng et al., “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3656–3663.

[5] Z. Pan et al., “Spatio-temporal meta learning for urban traffic prediction,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1462–1476, Mar. 2022.

[6] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng, “Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2348–2359, May 2022.

[7] Y. Gong, Z. Li, J. Zhang, W. Liu, and Y. Zheng, “Online spatio-temporal crowd flow distribution prediction for complex metro system,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 865–880, Feb. 2022.

[8] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, “Modeling extreme events in time series prediction,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1114–1122.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[10] J. Deng, X. Chen, Z. Fan, R. Jiang, X. Song, and I. W. Tsang, “Do RNN and LSTM have long memory?,” in *Neural Inf. Process. Syst.* Conf. Knowl. Discov. Data Mining, 2019, pp. 3656–3663.

[11] R. Jiang et al., “DeepCrowd: A deep model for large-scale city-wide crowd density and flow prediction,” *IEEE Trans. Knowl. Data Eng.*, early access, May 03, 2021, doi: 10.1109/TKDE.2021.3077056.

[12] J. Deng, X. Chen, Z. Fan, R. Jiang, X. Song, and I. W. Tsang, “The pulse of urban transport: Exploring the co-evolving pattern for spatio-temporal forecasting,” *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 6, pp. 1–25, 2021.

[13] R. Jiang et al., “DeepUrbanEvent: A system for predicting city-wide crowd dynamics at big events,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2114–2122.

[14] X. Geng et al., “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3656–3663.

[15] Z. Pan et al., “Spatio-temporal meta learning for urban traffic prediction,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1462–1476, Mar. 2022.

[16] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng, “Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2348–2359, May 2022.

[17] Y. Gong, Z. Li, J. Zhang, W. Liu, and Y. Zheng, “Online spatio-temporal crowd flow distribution prediction for complex metro system,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 865–880, Feb. 2022.

[18] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, “Modeling extreme events in time series prediction,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1114–1122.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[20] J. Deng, X. Chen, Z. Fan, R. Jiang, X. Song, and I. W. Tsang, “Do RNN and LSTM have long memory?,” in *Neural Inf. Process. Syst.* Conf. Knowl. Discov. Data Mining, 2019, pp. 3656–3663.
[19] Y.-Y. Chang, F.-Y. Sun, Y.-H. Wu, and S.-D. Lin, “A memory-network based solution for multivariate time-series forecasting,” 2018, arXiv:1809.02105.

[20] X. Tang, H. Yao, Y. Sun, C. C. Aggarwal, P. Mitra, and S. Wang, “Joint modeling of local and global temporal dynamics for multivariate time-series forecasting with missing values,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 9565–9573.

[21] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, “Modeling long-and short-term temporal patterns with deep neural networks,” in Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2018, pp. 95–104.

[22] Q. Tan et al., “DATA-GRU: Dual-attention time-aware gated recurrent unit for irregular multivariate time series,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 9951–9958.

[23] C. Fan et al., “Multi-horizon time series forecasting with temporal attention learning,” in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2019, pp. 2527–2535.

[24] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural In. Process. Syst., 2017, pp. 5998–6008.

[25] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph WaveNet for deep spatial-temporal graph modeling,” in Proc. 28th Int. Joint Conf. Artif. Intell., 2019, pp. 1907–1913.

[26] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, “Connecting the dots: Multivariate time series forecasting with graph neural networks,” 2020, arXiv:2005.11650.

[27] R. Shen, H.-F. Yu, and I. S. Dhillon, “Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 4837–4846.

[28] H. Zhou et al., “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in Proc. AAAI Conf. Artif. Intell., 2021, pp. 10 116–11 115.

[29] J. Cheng, K. Huang, and Z. Zheng, “Towards better forecasting by fusing near and distant future visions,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 3593–3600.

[30] S. Rangan, V. Flunkert, J. Gasthaus, and T. Januschowski, “DeepAR: Probabilistic forecasting with autoregressive recurrent networks,” Int. J. Forecasting, vol. 36, pp. 1181–1191, 2019.

[31] H.-F. Yu, N. Rao, and I. S. Dhillon, “Temporal regularized matrix factorization for high-dimensional time series prediction,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 847–855.

[32] R. Yu, S. Zheng, A. Anandkumar, and Y. Yue, “Long-term forecasting using higher order tensor KNNs,” 2017, arXiv:1711.00073.

[33] Y. Yu, S. Kang, K. Jang, X. Yi, and Y. Zheng, “GEOMAN: Multilevel attention networks for geo-sensory time series prediction,” in Proc. Int. Joint Conf. Artif. Intell., 2018, pp. 3428–3434.

[34] G. Spadon, S. Hong, B. Brandoli, S. Matwin, J. F. Rodrigues-Jr, and J. Sun, “Pay attention to evolution: Time series forecasting with deep graph-evolution learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 9, pp. 5368–5384, Sep. 2022.

[35] D. Cao et al., “Spectral-temporal graph neural network for multivariate time-series forecasting,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 17 766–17 778.

[36] S. Huang, D. Wang, X. Wu, and A. Tang, “DSANet: Dual self-attention network for multivariate time series forecasting,” in Proc. 28th ACM Int. Conf. Inf. Knowl. Manage., 2019, pp. 2129–2132.

[37] R. Jayasuriya, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, “Deep state space models for time series forecasting,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 7785–7794.

[38] Y. Wang, A. Smola, D. C. Maddix, J. Gasthaus, D. Foster, and T. Januschowski, “Deep factors for forecasting,” 2019, arXiv:1905.12417.

[39] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, “Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling,” in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2019, pp. 1227–1235.

[40] V. Le Guen and N. Thome, “Shape and time distortion loss for training deep time series forecasting models,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 4189–4201.

[41] T. Guo, T. Lin, and N. Antulov-Fantulin, “Exploring interpretable LSTM neural networks over multi-variable data,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 2494–2504.

[42] M. Binkowski, G. Marti, and P. Donnat, “Autoregressive convolutional neural networks for asynchronous time series,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 580–589.

[43] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015, arXiv:1502.03167.

[44] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” 2016, arXiv:1607.08022.

[45] Y. Wu and K. He, “Group normalization,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 3–19.

[46] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

[47] B. Li, F. Wu, K. Q. Weinberger, and S. Belongie, “Positional normalization,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 1622–1630.

[48] J. Lin and J. Liu, “Revisit batch normalization: New understanding and refinement via composition optimization,” in Proc. 22nd Int. Conf. Artif. Intell. Statist., 2019, pp. 3254–3263.

[49] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, “Understanding batch normalization,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 7694–7705.

[50] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 2483–2493.

[51] H. Daneshmand, J. Kohler, F. Bach, T. Hofmann, and A. Lucchi, “Batch normalization provably avoids rank collapse for randomly initialised deep networks,” 2020, arXiv:2003.01652.

[52] L. Huang, L. Zhao, Y. Zhou, F. Zhu, L. Liu, and L. Shao, “An investigation into the stochasticity of batch whitening,” in Proc. Int. Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6439–6448.

[53] A. Kessy, A. Lewin, and K. Strimmer, “Optimal whitening and decorrelation,” Amer. Statistician, vol. 72, no. 4, pp. 309–314, 2018.

[54] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[55] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, “Adaptive graph convolutional recurrent network for traffic forecasting,” Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 17 804–17 813.

Jinliang Deng received the BS degree in computer science from Peking University in 2017, and the MS degree in computer science from The Hong Kong University of Science and Technology in 2019. He is currently working toward the PhD degree with the Australian Artificial Intelligence Institute, University of Technology Sydney and the Department of Computer Science and Engineering, Southern University of Science and Technology. His research interests include time series forecasting, urban computing and deep learning.

Xiushi Chen received the BS and MS degrees in computer science from Peking University in 2015 and 2018, respectively. He is currently working toward the PhD degree with the Department of Computer Science, University of California, Los Angeles. His research interests include natural language processing, knowledge graph, neural machine reasoning and reinforcement learning.

Renchong Jiang received the BS degree in software engineering from the Dalian University of Technology, China, in 2012, the MS degree in information science from Nagoya University, Japan, in 2015, and the PhD degree in civil engineering from The University of Tokyo, Japan, in 2019. From 2019, he has been an assistant professor with the Information Technology Center, The University of Tokyo.
Xuan Song received the PhD degree in signal and information processing from Peking University in 2010. In 2017, he was selected as Excellent Young Researcher of Japan MEXT. He has served as an associate editor, guest editor, area chair, senior program committee member for many prestigious journals and top-tier conferences, such as IMWUT, IEEE Transactions on Multimedia, WWW Journal, ACM Transactions on Intelligent Systems and Technology, IEEE Transactions on Knowledge and Data Engineering, Big Data Journal, UbiComp, IJCAI, AAAI, ICCV, CVPR etc. His main research interests are AI and its related research areas, such as data mining and urban computing. To date, he has published more than 100 technical publications in journals, book chapters, and international conference proceedings, including more than 60 high-impact papers in top-tier publications for computer science. His research has been featured in many Chinese, Japanese and international venues, including the United Nations, the Discovery Channel, and Fast Company Magazine. He received the Honorable Mention Award at UbiComp 2015.

Ivor W. Tsang (Fellow, IEEE) is an ARC future fellow and professor of Artificial Intelligence with the University of Technology Sydney (UTS), Australia. He is also the research director of the Australian Artificial Intelligence Institute. His research interests include transfer learning, generative models, and Big Data analytics for data with extremely high dimensions. In 2013, he received his prestigious ARC Future Fellowship for his research regarding machine learning on Big Data. In 2019, his JMLR paper titled “Towards ultrahigh dimensional feature selection for Big Data” received the International Consortium of Chinese Mathematicians Best Paper Award. In 2020, he was recognized as the AI 2000 AAAI/IJCAI Most Influential Scholar in Australia for his outstanding contributions to the field of AAAI/IJCAI between 2009 and 2019. His research on transfer learning granted him the Best Student Paper Award for CVPR 2010 and 2014 IEEE Transactions on Multimedia Prize Paper Award. In addition, he received the IEEE TNN Outstanding 2004 Paper Award in 2007. He serves as a senior area chair/area chair for NeurIPS, ICML, AISTATS, AAAI and IJCAI, and the Editorial Board for Journal of Machine Learning Research, MLJ, and IEEE Transactions on Pattern Analysis and Machine Intelligence.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.