Physics-Informed Long-Sequence Forecasting From Multi-Resolution Spatiotemporal Data

Chuizheng Meng[1] , **Hao Niu**[2] , **Guillaume Habault**[2] , **Roberto Legaspi**[2] , **Shinya Wada**[2] , **Chihiro Ono**[2] , **Yan Liu**[1]
[1]University of Southern California
[2]KDDI Research, Inc.
chuizhem@usc.edu, {ha-niu,gu-habault,ro-legaspi,sh-wada,ono}@kddi-research.jp, yanliu.cs@usc.edu

Abstract

Spatiotemporal data aggregated over regions or time windows at various resolutions demonstrate heterogeneous patterns and dynamics in each resolution. Meanwhile, the multi-resolution characteristic provides rich contextual information, which is critical for effective long-sequence forecasting. The importance of such inter-resolution information is more significant in practical cases, where fine-grained data is usually collected via approaches with lower costs but also lower qualities compared to those for coarse-grained data. However, existing works focus on uni-resolution data and cannot be directly applied to fully utilize the aforementioned extra information in multi-resolution data. In this work, we propose Spatiotemporal Koopman Multi-Resolution Network (ST-KMRN), a physics-informed learning framework for long-sequence forecasting from multi-resolution spatiotemporal data. Our method jointly models data aggregated in multiple resolutions and captures the inter-resolution dynamics with the self-attention mechanism. We also propose downsampling and upsampling modules among resolutions to further strengthen the connections among data of multiple resolutions. Moreover, we enhance the modeling of intra-resolution dynamics with physics-informed modules based on Koopman theory. Experimental results demonstrate that our proposed approach achieves the best performance on the long-sequence forecasting tasks compared to baselines without a specific design for multi-resolution data.

1 Introduction

Forecasting from spatiotemporal data has wide applications in domains such as transportation and energy. The input and output of such forecasting tasks are both sequences of graph signal frames, where each frame is a graph with multivariate features defined on nodes and edges. In these applications, long-run forecasting is usually required for planning and policy making. As illustrated by [Zhou *et al.*, 2021], forecasting long sequences into the long-run future has higher requirements on the long-range alignment ability of models compared to short-run forecasting tasks.

One key aspect for forecasting long sequences is effectively modeling the complex patterns and dynamics of real-world spatiotemporal data aggregated in various spatial and temporal resolutions. Figure 1 shows an example from the taxi demand dataset. Data aggregated in different spatial (first 4 rows vs. the last row) and temporal (left/middle/right columns) resolutions demonstrate correlated but heterogeneous patterns, implying the necessity of refined modeling of multi-resolution data. In practical cases where data of high resolutions usually suffer from high missing rates and low signal-to-noise ratios due to the high cost of collection, correctly capturing the interaction among data in various resolutions is even more critical for forecasting.

While existing works have achieved great success in short-run predictions of uni-resolution spatiotemporal data within several hours into the future [Zhang *et al.*, 2017; Geng *et al.*, 2019; Yao *et al.*, 2019; Wu *et al.*, 2019; Zheng *et al.*, 2020; Chen *et al.*, 2020; Zhou *et al.*, 2020; Cao *et al.*, 2020], research on long-run forecasting of spatiotemporal data remains hardly developed. [Zhou *et al.*, 2021] proposes an efficient Transformer-based [Vaswani *et al.*, 2017] architecture for long-sequence time series forecasting, but still lacks modeling of spatial relations and ignores the inter-resolution dynamics. We include a more comprehensive summary of related works in the appendix[1].

To bridge the gap between multi-resolution spatiotemporal data and long-sequence forecasting, the proposed model must address challenges in two aspects: (1) *Inter-resolution modeling*. The model must be able to capture interactions of data among resolutions to fully utilize them for forecasting. (2) *Intra-resolution modeling*. The model should effectively extract information representing the dynamics from either short or long sequences of each resolution.

In this paper, we propose Spatiotemporal Koopman Multi-Resolution Network (ST-KMRN) to address both challenges. For better inter-resolution modeling, we leverage the self-attention mechanism to fuse and communicate representations of input data in all temporal resolutions. The model is then jointly trained with forecasting losses over all spatial and tem-

[1] https://github.com/mengcz13/mengcz13.github.io/raw/master/pdf/ijcai2022-appendix.pdf

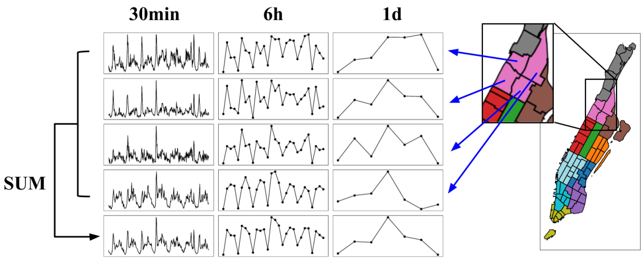


Figure 1: Example of multi-resolution spatiotemporal data: Taxi Pickup Rates. Each of the first 4 rows displays the change of taxi pickup rates (total pickup times in 30 minutes) over a one week period for one of the 4 pink taxi zones (bounded with black in the map).The last row shows the aggregated (summed) taxi pickup rates in the borough composed of the 4 pink taxi zones, and thus is of a coarse spatial resolution. Each column stands for one temporal aggregation resolution. Data in coarser spatial/temporal resolutions is aggregated from data in finer resolutions when the latter is fully observed. In other scenarios, it can also be collected from different data sources and thus have heterogeneous qualities.

poral resolutions. We further construct downsampling and upsampling modules among forecasting outcomes of different temporal resolutions to enhance inter-resolution connections. To improve intra-resolution modeling, we combine the Koopman theory-based modeling of dynamic systems with deep learning-based prediction modules for more principled modeling of dynamics. The inferred Koopman matrix offers interpretations of heterogeneous dynamics in different resolutions at the same time.

Our contributions are: (1) ST-KMRN captures inter-resolution dynamics of spatiotemporal data via connecting representations in different levels of resolutions via self-attention, providing an effective way of leveraging multi-resolution data. (2) ST-KMRN further improves inter-resolution modeling with upsampling and downsampling modules among predictions of various resolutions. (3) ST-KMRN improves the modeling of intra-resolution dynamics with the combination of Koopman theory-based modeling and deep learning-based forecasting models, which also provides interpretable information of different dynamics in multi-resolution data. (4) ST-KMRN achieves state-of-the-art performance on the long-sequence forecasting tasks from real-world spatiotemporal datasets.

2 Problem Formulation

In this work, we focus on the task of forecasting spatiotemporal multivariate time series with multiple resolutions of both space and time. We denote the set of regions by $\mathbb{S} = \{s_1, s_2, \dots, s_N\}$, the set of P historical time steps by $\mathbb{T}_H = \{t_1, t_2, \dots, t_P\}$, the set of Q future time steps by $\mathbb{T}_F = \{t_{P+1}, \dots, t_{P+Q}\}$, and the multivariate variable of dimension D at region s and time step t by $\boldsymbol{x}_{t,s} \in \mathbb{R}^D$.

Definition 1 (***Resolution***)**.** A resolution $\mathbb{R}(\mathbb{X})$ is a *partition* of the set $\mathbb{X}$. In addition, for a temporal resolution $\mathbb{R}(\mathbb{T})$ ($\mathbb{T} = \mathbb{T}_H$ or $\mathbb{T} = \mathbb{T}_F$), we require $|\mathbb{T}| = |\mathbb{R}(\mathbb{T})| \times r$, $\exists r \in \mathbb{N}$, and $\mathbb{R}(\mathbb{T}) = \{\{t_1, \dots, t_r\}, \dots, \{t_{(|\mathbb{R}(\mathbb{T})|-1)\times r+1}, \dots, t_{|\mathbb{R}(\mathbb{T})|\times r}\}\}$. We name r as the *scale* of the temporal resolution $\mathbb{R}(\mathbb{T})$.

Definition 2 (***Aggregated Variable***)**.** The aggregated multivariate variable over a set of regions $\mathbb{S}'$ and a set of time steps $\mathbb{T}'$ is defined as $\boldsymbol{x}^{\text{agg}}_{\mathbb{T}',\mathbb{S}'} = \text{agg}(\{\boldsymbol{x}_{t,s} \mid t \in \mathbb{T}', s \in \mathbb{S}'\})$, where agg is some aggregation function $\text{agg} : \mathbb{R}^{N\times D} \to \mathbb{R}^D$, such as summation or average. We omit agg in following notations since the aggregation function is usually the same for a given dataset and can be inferred from the context.

Definition 3 (***Observation at Given Spatial and Temporal Resolutions***)**.** The observation at a given spatial resolution $\mathbb{R}(\mathbb{S})$ and a given temporal resolution $\mathbb{R}(\mathbb{T})$ is defined as a set of aggregated multivariate variables: $\mathbb{O}_{\mathbb{R}(\mathbb{T}),\mathbb{R}(\mathbb{S})} = \{\boldsymbol{x}_{r_T,r_S} | r_T \in \mathbb{R}(\mathbb{T}), r_S \in \mathbb{R}(\mathbb{S})\} \in \mathbb{R}^{|\mathbb{R}(\mathbb{T})|\times|\mathbb{R}(\mathbb{S})|\times D}$.

We then formulate the input and output of the forecasting problem with multi-resolution spatiotemporal input data as follows:

Input (1) The set of regions $\mathbb{S}$, the set of historical time steps $\mathbb{T}_H$, and the set of future time steps $\mathbb{T}_F$. (2) The list of available temporal resolutions $\mathbf{TR} = \{\mathbb{R}^T_1(\mathbb{T}_H), \dots, \mathbb{R}^T_{R_T}(\mathbb{T}_H)\}$, and the list of available spatial resolutions $\mathbf{SR} = \{\mathbb{R}^S_1(\mathbb{S}), \dots, \mathbb{R}^S_{R_S}(\mathbb{S})\}$. $\mathbb{R}^H_1(\mathbb{T}_H), \mathbb{R}^H_1(\mathbb{S})$ are the highest temporal and spatial resolution respectively. In other words, $|r_t| = 1,\ \forall\, r_t \in \mathbb{R}^T_1(\mathbb{T}_H)$, and $|r_s| = 1,\ \forall\, r_s \in \mathbb{R}^S_1(\mathbb{S})$. (3) The set of historical spatio-temporal resolution pairs will be $\mathbf{HR} = \mathbf{TR} \times \mathbf{SR}$. (4) The set of historical observations in multiple spatial and temporal resolutions $\mathbf{O} = \{\mathbb{O}_{TR,SR} \mid (TR, SR) \in \mathbf{HR}\}$.

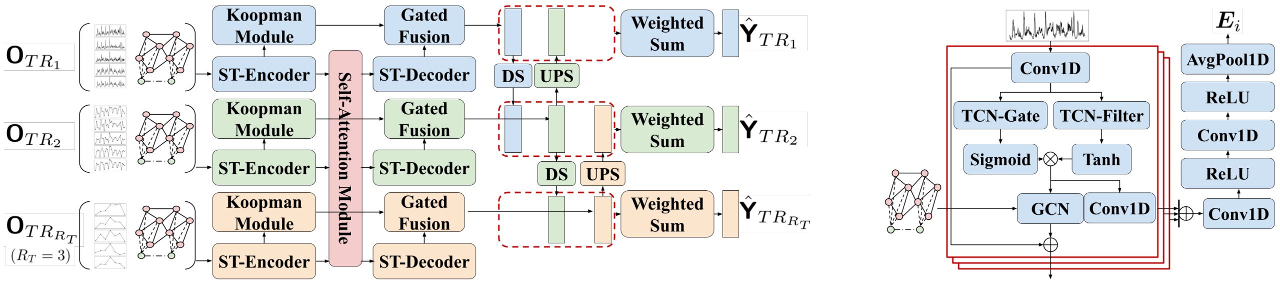
Output The set of forecast values at the target spatial and temporal resolution $\hat{\mathbf{Y}} = \mathbb{O}_{\mathbb{R}^T_{out}(\mathbb{T}_F),\mathbb{R}^S_{out}(\mathbb{S})}$.

3 Proposed Method: Spatiotemporal Koopman Multi-Resolution Network

In this section, we introduce our proposed model: Spatiotemporal Koopman Multi-Resolution Network (ST-KMRN). Figure 2a provides an overview of its architecture.

Input historical observations $\mathbf{O}$ in multiple spatial and temporal resolutions are first grouped by temporal resolutions into $\{\mathbf{O}_{TR_1}, \dots, \mathbf{O}_{TR_{R_T}}\}$, where $\mathbf{O}_{TR_i} = \text{concat}(\{\mathbb{O}_{TR_i,SR_1}, \dots, \mathbb{O}_{TR_i,SR_{R_S}}\}) \in \mathbb{R}^{|TR_i|\times N_S\times D}, N_S = \sum_{k=1}^{R_S} |SR_k|$ is the concatenated multivariate time series along the spatial dimension. For $\mathbf{O}_{TR_i}$, we construct a spatial hierarchical graph $G_i = (V_i, E_i)$, where $V_i = \bigcup_{k=1}^{R_S} SR_k$ contains region sets at all spatial resolution levels and E_i is composed of 2 types of edges: (1) *intra-spatial-resolution edges* constructed with prior knowledge such as spatial information or connection strengths; (2) *inter-spatial-resolution edges* between all pairs of region sets $(\mathbb{S}_p, \mathbb{S}_q)$ iff. $\mathbb{S}_p \subset \mathbb{S}_q$.

Each pair of $(\mathbf{O}_{TR_i}, G_i)$ is fed into the i-th spatiotemporal encoder to generate a temporal-resolution-specific embedding $\boldsymbol{E}_i \in \mathbb{R}^{N_S\times H}$, where H is the embedding dimension. Embeddings of all temporal resolutions are then propagated and updated with the self-attention module and the updated embedding $\boldsymbol{E}'_i$ serves as the input of the i-th decoder to get the



(a) Overview of ST-KMRN's architecture ($R_T = 3$).

(b) Neural network based module.

Figure 2: ST-KMRN's architecture.

deep learning based forecasting result $\hat{\mathbf{Y}}^{DL}_{TR_i}$. Meanwhile, a Koopman Theory based forecasting module takes $\mathbf{O}_{TR_i}$ as its input and gives its forecasting $\hat{\mathbf{Y}}^{K}_{TR_i}$ in parallel. $\hat{\mathbf{Y}}^{DL}_{TR_i}$ and $\hat{\mathbf{Y}}^{K}_{TR_i}$ are then fused with a gating mechanism to get the first-stage forecasting result $\hat{\mathbf{Y}}^{1}_{TR_i}$. We further design upsampling and downsampling modules for the second-stage forecasting: $\hat{\mathbf{Y}}^{1}_{TR_i}$, $\hat{\mathbf{Y}}^{ds}_{TR_i} = \text{DownSampling}_{i-1}(\hat{\mathbf{Y}}^{1}_{TR_{i-1}})$, $\hat{\mathbf{Y}}^{ups}_{TR_i} = \text{UpSampling}_{i+1}(\hat{\mathbf{Y}}^{1}_{TR_{i+1}})$ is combined with the attention mechanism for the final forecast $\hat{\mathbf{Y}}_{TR_i}$ (for $i = 1$ / N, $\hat{\mathbf{Y}}^{ds}_{TR_{i-1}}$ / $\hat{\mathbf{Y}}^{ups}_{TR_{i+1}}$ is omitted respectively).

3.1 Physics-Informed Modeling of Intra-Resolution Dynamics

In this part, we introduce our proposed physics-informed modeling of intra-resolution dynamics, where a neural network based module and a Koopman theory based module are combined for encoding and forecasting in each temporal resolution.

Neural Network Based Module (ST-Encoder and ST-Decoder)

The architecture of the neural network based module is a modification of Graph-WaveNet [Wu *et al.*, 2019]. Figure 2b shows the architecture of the encoder part, where multiple temporal convolutional layers (TCN) [Oord *et al.*, 2016] and graph convolutional layers (GCN) [Kipf and Welling, 2017] are stacked alternately to encode spatial dependencies at various temporal scales. The output of the encoder is an embedding vector $\boldsymbol{E}_i$ summarizing the input sequence $\mathbf{O}_{TR_i}$. $\boldsymbol{E}_i$ is then propagated with embeddings from other resolutions into $\boldsymbol{E}'_i$ via self-attention [Vaswani *et al.*, 2017]. The decoder is composed of two 1D convolutional layers applied along the spatial dimension, which maps $\boldsymbol{E}'_i$ to the forecasting result $\hat{\mathbf{Y}}^{DL}_{TR_i}$.

Koopman Module

Koopman Theory Given a non-linear dynamic system with its state vector at time t denoted as $\boldsymbol{s}_t \in \mathbb{R}^m$. The system can be described as $\boldsymbol{s}_{t+1} = F(\boldsymbol{s}_t)$. As defined in [Koopman, 1931], the Koopman operator $\mathcal{K}_F$ is a linear transformation defined on a function space $\mathcal{F}$ by $\mathcal{K}_F g = g \circ F$ for every $g : \mathbb{R}^m \to \mathbb{R}$ that belongs to the infinite-dimensional Hilbert space $\mathcal{F}$. With this definition, we have $\mathcal{K}_F g(\boldsymbol{s}_t) = g \circ F(\boldsymbol{s}_t) = g(\boldsymbol{s}_{t+1})$.

The Koopman theory [Koopman, 1931] guarantees the existence of $\mathcal{K}$, but in practice we often assume the existence of an invariant finite-dimensional subspace $\mathcal{G}$ of $\mathcal{F}$ spanned by k bases $\{g_1, g_2, \ldots, g_k\}$. Define $\boldsymbol{g}_t = [g_1(\boldsymbol{s}_t), g_2(\boldsymbol{s}_t), \ldots, g_k(\boldsymbol{s}_t)]^T$ and $\boldsymbol{g}_{t+1} = [g_1(\boldsymbol{s}_{t+1}), g_2(\boldsymbol{s}_{t+1}), \ldots, g_k(\boldsymbol{s}_{t+1})]^T$, under the assumption we have $\boldsymbol{g}_t, \boldsymbol{g}_{t+1} \in \mathcal{G}$ and there exists a Koopman matrix $\boldsymbol{K} \in \mathbb{R}^{k \times k}$ s.t. $\boldsymbol{g}_{t+1} = \boldsymbol{K}\boldsymbol{g}_t$.

Koopman Theory Based Modeling of Temporal Dynamics Compared to directly modeling the state propagation function F with neural networks, the Koopman theory provides prior knowledge related to the system. In addition, the resulting Koopman matrix is more interpretable as the propagation of hidden states can be represented as a linear mapping.

The key problem is to find the pair of mappings between the state space $\mathbb{R}^m$ and the invariant subspace $\mathcal{G} \in \mathbb{R}^k$: $\boldsymbol{g} : \mathbb{R}^m \to \mathbb{R}^k$ and $\boldsymbol{g}^{-1} : \mathbb{R}^k \to \mathbb{R}^m$. Here we model them with 2 multi-layer perceptrons (MLPs). $\boldsymbol{g}$ maps each frame of the input $\mathbf{O}_{TR_i}$ to the subspace, and generates $\boldsymbol{H}_{TR_i} \in \mathbb{R}^{|TR_i| \times k}$. Then, the Koopman matrix $\boldsymbol{K}_{TR_i}$ is estimated as

$$\arg\min_{\boldsymbol{K}} \left\| \boldsymbol{K}(\boldsymbol{H}_{TR_i}[0:-1])^T - (\boldsymbol{H}_{TR_i}[1:])^T \right\|_2^2 . \qquad (1)$$

With $\boldsymbol{H}_{TR_i}[-1]$ as the starting state $\boldsymbol{q}_0$ for forecasting, the module forecasts T_i steps into the future in a recurrent way: $\hat{\mathbf{Y}}^{K}_{TR_i} = [\boldsymbol{g}^{-1}(\boldsymbol{K}_{TR_i}\boldsymbol{q}_0), \boldsymbol{g}^{-1}(\boldsymbol{K}^2_{TR_i}\boldsymbol{q}_0), \ldots, \boldsymbol{g}^{-1}(\boldsymbol{K}^{T_i}_{TR_i}\boldsymbol{q}_0)]$.

Gated Fusion of ST-Encoder and Koopman Module

While the Koopman theory based module is more principled and can be incorporated with prior knowledge of the system, the neural network module is far more expressive and flexible. To combine the advantages of both modules, we design a fusion module that jointly leverages their forecasting. More specifically, we use a 2-layer 1D convolutional neural network (CNN) with the sigmoid activation function to generate a vector. The vector controls the ratio of each module's prediction in the combined forecasting results as follows:

$$\boldsymbol{\eta}_{TR_i} = \boldsymbol{f}_{\text{Gate}}(\text{concat}(\hat{\mathbf{Y}}^{DL}_{TR_i}, \hat{\mathbf{Y}}^{K}_{TR_i})) \qquad (2)$$

$$\hat{\mathbf{Y}}^{1}_{TR_i} = (1 - \boldsymbol{\eta}_{TR_i}) \otimes \hat{\mathbf{Y}}^{DL}_{TR_i} + \boldsymbol{\eta}_{TR_i} \otimes \hat{\mathbf{Y}}^{K}_{TR_i}, \qquad (3)$$

where $\otimes$ is the element-wise product.

3.2 Inter-Resolution Dynamics Modeling

In this part, we introduce the self-attention module that fuses representations over all temporal resolutions as well as the upsampling and downsampling modules for the second stage forecasting.

Self-Attention Module

The self-attention mechanism, proposed in [Vaswani *et al.*, 2017], captures dependencies of input elements without regard to their distance in the input sequence. With similar motivation, we adopt self-attention to model the interaction of representations from multiple temporal resolutions. Given the list of resolution-specific embeddings $[\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_{R_T}]$, we concatenate all embeddings to construct the input representation as $\boldsymbol{E} \in \mathbb{R}^{N_S \times R_T \times H}$. We then update the embeddings of all regions in parallel via multi-head self-attention across resolutions.

Upsampling and Downsampling Modules

Upsampling Module We set up a learnable upsampling module $\mathrm{UpSampling}_i$ for each $\hat{\mathbf{Y}}^{1}_{TR_i}, i > 1$. Denote r_i as the scale of the temporal resolution TR_i. In our experimental settings we always have $r_i = k_i r_{i-1}$, $\exists k_i \in \mathbb{N}$, $\forall i > 1$. We implement the i-th upsampling module as a 1D CNN with D input channels and $k_i D$ output channels. Its output is then rearranged as the upsampled prediction in the same way as PixelShuffle [Shi *et al.*, 2016] for the upsampled forecasting results $\hat{\mathbf{Y}}^{ups}_{TR_{i-1}}$.

Downsampling Module We also set up a downsampling module $\mathrm{DownSampling}_i$ for each $\hat{\mathbf{Y}}^{1}_{TR_i}, i < R_T$. In the same experimental settings as mentioned above, we implement the downsampling module as an aggregation operator, which is of the same type of aggregation for generating data in coarser temporal resolutions. For the temporal resolution $TR_i, i < R_T$, the operator aggregates the prediction in $\hat{\mathbf{Y}}^{1}_{TR_i}$ every r_{i+1}/r_i steps as the downsampled prediction $\hat{\mathbf{Y}}^{ds}_{TR_{i+1}}$.

Weighted Summation of Forecasting Results For the i-th temporal resolution ($1 < i < R_T$), we use the weighted summation of the first-stage results, the upsampled results, and the downsampled results as the second-stage prediction. Weights for summation are the output of a learnable module implemented as CNN with softmax activation function defined for the i-th resolution:

$$\boldsymbol{w}_{TR_i} = \boldsymbol{f}_{weight}(\mathrm{concat}(\hat{\mathbf{Y}}^{1}_{TR_i}, \hat{\mathbf{Y}}^{ups}_{TR_i}, \hat{\mathbf{Y}}^{ds}_{TR_i})) \tag{4}$$

$$\begin{aligned} \hat{\mathbf{Y}}_{TR_i} = \; & \boldsymbol{w}_{TR_i}[:,:,0] \otimes \hat{\mathbf{Y}}^{1}_{TR_i} + \boldsymbol{w}_{TR_i}[:,:,1] \otimes \hat{\mathbf{Y}}^{ups}_{TR_i} \\ & + \boldsymbol{w}_{TR_i}[:,:,2] \otimes \hat{\mathbf{Y}}^{ds}_{TR_i}, \end{aligned} \tag{5}$$

where $\otimes$ is the element-wise product. For cases where $i = 1$[resp. R_T], the weighted summation is calculated in the same way but with the $\hat{\mathbf{Y}}^{ds}_{TR_i}$ [resp. $\hat{\mathbf{Y}}^{ups}_{TR_i}$] term removed.

3.3 Loss Function

Denote the forecasting loss function as $\mathcal{L}(\hat{\mathbf{Y}}_{pred}, \mathbf{Y}_{true})$, the total loss for training the model is

$$\mathcal{L}_{total} = \sum_{i=1}^{R_T} \mathcal{L}_{TR_i}, \quad \text{where} \tag{6}$$

$$\mathcal{L}_{TR_i} = \sum_{\hat{\mathbf{Y}} \in \left\{ \begin{array}{c} \hat{\mathbf{Y}}^{DL}_{TR_i}, \hat{\mathbf{Y}}^{K}_{TR_i}, \\ \hat{\mathbf{Y}}^{1}_{TR_i}, \hat{\mathbf{Y}}^{ups}_{TR_i}, \\ \hat{\mathbf{Y}}^{ds}_{TR_i}, \hat{\mathbf{Y}}_{TR_i} \end{array} \right\}} \mathcal{L}(\hat{\mathbf{Y}}, \mathbf{Y}_{TR_i}) + \mathcal{L}^{(\mathrm{KM})}_{TR_i} \tag{7}$$

$$\begin{aligned} \mathcal{L}^{(\mathrm{KM})}_{TR_i} = \; & \textstyle\sum_{1 \leqslant p,q \leqslant |TR_i|} \big| |\boldsymbol{g}(\mathbf{Y}_{TR_i}[p]) - \boldsymbol{g}(\mathbf{Y}_{TR_i}[q])| \\ & - |\mathbf{Y}_{TR_i}[p] - \mathbf{Y}_{TR_i}[q]| \big| \, . \end{aligned} \tag{8}$$

$\mathcal{L}^{(\mathrm{KM})}_{TR_i}$ encourages the mapping from the state space to the invariant subspace in the Koopman module to preserve the distance.

4 Experiments

Datasets We evaluate the performance of ST-KMRN and all baselines on 3 datasets: (1) New York Yellow Taxi Trip Record Data (**YellowCab**) [NYCTLC, 2021] in 2017-2019; (2) New York Green Taxi Trip Record Data (**GreenCab**) [NYCTLC, 2021] in 2017-2019; and (3) Solar Energy Data (**Solar Energy**) [NREL, 2021] of Alabama in 2006. We use sliding windows to generate input/output sequence pairs ordered by starting time and divide all pairs into train/validation/test sets with the ratio 60%/20%/20%. Since all datasets are originally in a single resolution, we construct one extra spatial resolution (named as "agg regions") and two extra temporal resolutions on all datasets by aggregation to simulate the multi-resolution scenario.We list statistics and task settings of each dataset in the appendix.

Baselines We compare our model ST-KMRN with the following baselines: (1) Historical Averaging (**HA**): We use the averaged value of historical frames with the 1-week period as the prediction. (2) **Static**: We use the value from the last available frame in the input sequence with the 1-week period as the prediction. (3) Gated Recurrent Unit (**GRU**) [Chung *et al.*, 2014] (4) **Informer** [Zhou *et al.*, 2021] (5) **Graph WaveNet** [Wu *et al.*, 2019] (6) **MTGNN** [Wu *et al.*, 2020] (7) **KoopmanAE** [Azencot *et al.*, 2020]. For all baselines, we concatenate input in multiple temporal resolutions as features into one single temporal resolution. We have also considered other recent works on short-term spatiotemporal forcasting as baselines, such as AGCRN [Bai *et al.*, 2020], GMAN [Zheng *et al.*, 2020], DGCRN [Li *et al.*, 2021]. However, their designs induce high computation and memory complexity on long sequential data and prohibit us from retrieving results practically.

4.1 Long-Sequence Forecasting With Fully Observed Input

Set-up We train all models and evaluate their forecasting performance within certain horizons into the future. We repeat

Data	YellowCab		GreenCab		Solar Energy	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	21.934	35.473	3.764	5.701	*69.401*	*152.590*
Static	*14.004*	*26.324*	2.076	3.080	71.758	179.680
GRU	25.625(0.188)	38.207(0.180)	2.668(0.056)	3.894(0.082)	114.667(9.018)	201.622(10.462)
Informer	22.210(2.046)	33.484(2.722)	1.926(0.062)	2.818(0.079)	81.456(5.181)	171.407(10.061)
Graph WaveNet	16.889(0.115)	30.929(0.257)	*1.801(0.011)*	*2.787(0.019)*	135.644(0.062)	290.471(0.186)
MTGNN	18.914(0.592)	34.017(0.985)	2.222(0.022)	3.462(0.009)	130.357(3.472)	283.506(5.695)
KoopmanAE	16.922(0.680)	28.115(0.891)	2.612(0.131)	3.832(0.196)	148.614(2.297)	253.097(0.421)
ST-KMRN	**13.631(0.812)**	**24.090(1.054)**	**1.682(0.004)**	**2.479(0.017)**	**67.718(1.620)**	**148.304(2.279)**
RelErr	-2.7%	-8.5%	-6.6%	-11.1%	-2.4%	-2.8%
RelErrGW	-19.3%	-22.1%	-6.6%	-11.1%	-50.1%	-48.9%

Table 1: Forecasting results within the maximum possible horizon from fully observed input sequences. The lowest error is marked in bold and the second-lowest error in italic with underline. The row ”RelErr” shows the relative error change to the best baseline model and ”RelErrGW” to the Graph WaveNet baseline.

each experiment 3 times and report mean values and standard deviations of all metrics.

Discussion Table 1 shows the forecasting performance within the maximum possible horizon (10-day for YellowCab/GreenCab and 3-day for Solar Energy) of baselines and our proposed ST-KMRN measured in Mean Absolute Error (MAE) and Rooted Mean Squared Error (RMSE). Results within other horizons are in the appendix and we have similar observations. We observe that: (1) On all of 3 datasets, ST-KMRN outperforms all baselines, demonstrating the advantage of ST-KMRN in long-sequence forecasting with multi-resolution data. (2) ST-KMRN achieves significantly decreased forecasting errors compared to the Graph-WaveNet baseline, with whom its encoders and decoders share similar neural network architectures. The results demonstrate that the gain of ST-KMRN comes from the enhanced modeling of both intra-resolution and inter-resolution dynamics. (3) On the YellowCab and Solar Energy datasets, our proposed ST-KMRN still holds the best performance compared to other baselines, but its advantages over HA and Static are not as large as on the GreenCab dataset. This is due to the strong periodicity within the fully observed input sequences. We further conduct experiments in a more challenging setting: input data is partially observed, where the periodicity in input is weaker.

4.2 Long-Sequence Forecasting Results With Partially Observed Input

Set-up We evaluate the effect of missing data in input sequences by training and evaluating all models with varying ratios of observable frames in the input of the finest spatial and temporal resolution (named as “obs ratio”). This setting aims to simulate the practical scenario when the model needs to forecast with low-quality input data of high resolutions.

Discussion Table 2 shows the 10-day forecasting performance of all models with various *obs ratios* on YellowCab. When the input data suffers from high missing ratios, the periodicity of data is less beneficial for forecasting, and the capability of capturing non-periodic patterns becomes more critical. Thus, trainable models start having advantages over the Static baseline. Under all *obs ratios* we select (0.8/0.6/0.4/0.2), our proposed ST-KMRN achieves the lowest forecasting errors, especially for higher ones (0.4 and 0.2), demonstrating its advantages with partially observed input. We also provide results for GreenCab and Solar Energy in the appendix, from which we have similar observations.

4.3 Ablation Study

Set-up We conduct the ablation study on the YellowCab dataset with 80% observation ratio to evaluate the contribution of proposed components. By removing each component from ST-KMRN, we have the following settings: (1) **w/o Self-Attn**: ST-KMRN without the self-attention module. (2) **w/o Koopman**: ST-KMRN without the Koopman module. The prediction will only be performed from the neural network module. (3) **w/o ups/ds**: ST-KMRN without upsampling and downsampling modules. Only the decoder of the target resolution will forecast. We repeat experiments of each setting for 3 times and report results in Table 3. Higher increase of errors after removing one module indicates larger contributions.

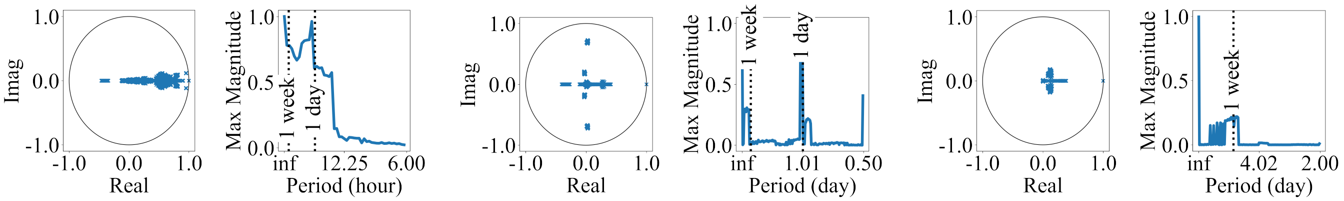
Discussion From Table 3, we observe that (1) the self-attention module applied on representations of different resolutions significantly improves performance. It allows information exchange among resolutions, and it contributes most on shorter horizons (6h). (2) The combination of both Koopman and neural network modules brings significant boosts to the forecasting performance for both short (6h) and long (10d) prediction horizons. It captures the correct patterns for each temporal resolution, which we will detail in Section ”Interpretability of Koopman Module”. (3) The upsampling and downsampling modules can improve performance on all forecasting horizons as (a) the downsampling module regularizes the output by minimizing the difference between its aggregated values with ground truth values in lower resolutions; (b) the upsampling module is trained to predict local temporal patterns and can fix the errors produced by the decoder, which forecasts values on all steps at once.

4.4 Interpretability of Koopman Module: Revealing Dynamics in Each Resolution

Set-up Since the Koopman module models the temporal dynamics as a linear system in the hidden space, the derived

Obs Ratio	0.8		0.6		0.4		0.2	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	21.95	35.52	21.97	35.58	21.99	35.62	22.08	35.82
Static	*14.21*	*26.66*	*15.77*	*29.73*	22.49	40.69	41.19	61.80
GRU	22.83(1.15)	34.97(1.62)	27.74(0.40)	41.21(0.46)	27.75(0.40)	41.27(0.45)	27.78(0.42)	41.31(0.46)
Informer	19.04(0.98)	29.21(1.29)	23.83(1.93)	35.35(2.50)	22.90(0.35)	34.31(0.50)	23.30(1.53)	34.66(1.96)
Graph WaveNet	16.53(0.29)	31.10(0.43)	17.08(0.30)	31.01(0.39)	*16.74(0.21)*	30.76(0.35)	*16.75(0.33)*	30.72(0.16)
MTGNN	18.43(0.88)	32.73(2.02)	19.41(0.40)	34.88(0.71)	19.81(0.09)	35.49(0.11)	18.95(0.11)	33.46(0.66)
KoopmanAE	18.43(1.57)	29.79(2.20)	19.90(1.47)	31.58(1.94)	18.96(2.08)	*30.09(2.99)*	19.07(0.58)	*30.66(0.77)*
ST-KMRN	**12.56(0.66)**	**22.58(0.71)**	**11.66(0.36)**	**21.46(0.58)**	**11.38(0.28)**	**21.17(0.45)**	**11.76(0.31)**	**21.64(0.48)**
RelErr	-11.6%	-15.3%	-26.1%	-27.8%	-32.0%	-29.6%	-29.8%	-29.4%
RelErrGW	-24.0%	-27.4%	-31.7%	-30.8%	-32.0%	-31.2%	-29.8%	-29.6%

Table 2: Forecasting results with partially observed input (YellowCab, Horizon=10d).



(a) 30-min (b) 6-hour (c) 1-day

Figure 3: Eigenvalues of Koopman matrices in the learned hidden space of input sequences in different resolutions. For each subfigure (a)(b)(c), the left part shows the distribution of eigenvalues on the complex plane with the unit circle, and the right part displays the distribution of maximum magnitudes of complex eigenvalues w.r.t periods ($\text{period} = 2\pi/\text{angle} \times \text{timewindow}$).

Horizon	6h		10d	
Metric	MAE	RMSE	MAE	RMSE
w/o Self-Attn	11.23(2.09) +58.84%	15.76(2.12) +26.38%	14.41(2.36) +14.73%	23.64(2.16) +4.69%
w/o Koopman	8.70(0.59) +23.06%	18.63(0.81) +49.40%	14.67(0.60) +16.80%	26.42(0.81) +17.01%
w/o ups/ds	7.94(0.51) +12.31%	12.87(0.52) +3.21%	13.45(0.20) +7.09%	23.18(0.12) +2.66%
ST-KMRN	**7.07(0.64)**	**12.47(0.58)**	**12.56(0.66)**	**22.58(0.71)**

Table 3: Ablation study results on YellowCab (Obs Ratio = 0.8) with the relative change of errors after removing each component.

Koopman matrix describing the system can provide rich interpretable information. With eigen decomposition applied to the Koopman matrix, we can decompose the system dynamics into components with different magnitudes and periods. An eigenvalue $\lambda e^{j\theta}$ corresponds to a component $f(t) = \lambda^t e^{j\theta t}$ with magnitude λ^t and frequency θ (i.e. period $T = \frac{2\pi}{\theta}$).

Discussion Figure 3 displays the distribution of eigenvalues of Koopman matrices for the input data of each resolution. We observe that: (1) For data in the resolution of 30-min, 6-hour, and 1-day, the maximum of magnitudes of eigenvalues reaches its peak value around the period of 24-hour, 1-day, and 7-day respectively (we ignore the eigenvalues with zero angles since they represent components with infinite periods and thus are non-periodic). This matches our observation that the taxi demand data shows strong daily and weekly patterns. (2) For data in the resolution of 30-min and 6-hour, we can still notice eigenvalues with large magnitudes around the 1-week period, but it is hard to distinguish them from other components including non-periodic ones. The reason is that in data with high temporal resolutions, long periods correspond to large numbers of time steps T and frequency values θ close to 0. When we switch to data with a lower temporal resolution (1-day), the pattern of the same period can be represented by components with larger distances to non-periodic components in the spectral domain and are easier to identify. This again implies the necessity of utilizing multi-resolution data for better modeling temporal patterns with various periods.

5 Conclusion

We propose Spatiotemporal Koopman Multi-Resolution Network (ST-KMRN), which boosts the long-sequence forecasting with enhanced modeling of inter-resolution and intra-resolution dynamics in multi-resolution data. To accomplish this, we introduce self-attention among representations of different resolutions together with upsampling and downsampling modules for better utilization of inter-resolution contextual information. Meanwhile, we improve the modeling of intra-resolution dynamics with the combination of Koopman theory based modeling. ST-KMRN achieves state-of-the-art performance on the long-sequence forecasting tasks from multiple real-world spatiotemporal datasets. Limitations of our work include that the available resolutions used by the model are fixed and are usually dependent on the task and data availability, and the model only applies to data with regular time steps. Our future works include extending the framework to find optimal data resolutions to guide the data collection process and enabling it to model and forecast irregular spatiotemporal sequences.

Acknowledgements

This work is supported by KDDI Research, Inc., granted to co-author Yan Liu in her academic role at the University of Southern California. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency.

References

[Azencot *et al.*, 2020] Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning*, pages 475–485. PMLR, 2020.

[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, 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, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, pages 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.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Geng *et al.*, 2019] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3656–3663, 2019.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Koopman, 1931] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the national academy of sciences of the united states of america*, 17(5):315, 1931.

[Li *et al.*, 2021] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2021.

[NREL, 2021] NREL. https://www.nrel.gov/grid/solar-power-data.html, 2021. Accessed: 2021-10-01.

[NYCTLC, 2021] NYCTLC. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page, 2021. Accessed: 2021-10-01.

[Oord *et al.*, 2016] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[Shi *et al.*, 2016] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

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

[Wu *et al.*, 2020] 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*, KDD '20, page 753–763, New York, NY, USA, 2020. Association for Computing Machinery.

[Yao *et al.*, 2019] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5668–5675, 2019.

[Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[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.*, 2020] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Hengchang Liu. Riskoracle: A minute-level citywide traffic accident forecasting framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1258–1265, 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 *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, page online. AAAI Press, 2021.