[Technical Appendix] Physics-Informed Long-Sequence Spatiotemporal Forecasting with Multi-Resolution 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

Related Works

Our work involves elements from forecasting from spatiotemporal data, long-term time series forecasting, and physics-informed modeling of spatiotemporal data. In the following subsections, we discuss related works in each domain in relation to our work.

Forecasting From Spatiotemporal Data

In recent years, methods based on Graph Neural Networks (GNNs) have shown superior performance in forecasting tasks from spatiotemporal data in various domains including physical simulation [Kipf *et al.*, 2018; Huang *et al.*, 2020], traffic [Li *et al.*, 2018; Wu *et al.*, 2019; Wu *et al.*, 2020; Yu *et al.*, 2018; Zheng *et al.*, 2020; Cao *et al.*, 2020], human motion [Yan *et al.*, 2018; Kipf *et al.*, 2018; Huang *et al.*, 2020], and climate [Seo *et al.*, 2020; Iakovlev *et al.*, 2020]. GNN-based methods typically utilize recurrent neural networks or one-dimensional convolution neural networks to capture temporal dependencies, and message passing or graph convolutions to model spatial dependencies. While existing works achieve state-of-the-art performance on spatiotemporal data with a single resolution, they lack the modeling of multi-resolution data. Although data in multiple resolutions can be processed as extra input features, it is not optimal for fully exploiting the rich contextual information among resolutions.

Long-Term Time Series Forecasting

One accompanying problem with multi-resolution forecasting is long-term time series forecasting as multi-step predictions for coarse temporal resolutions involve long prediction horizons into the future. [Zhou *et al.*, 2021] proposes an efficient Transformer-based method for modeling long time series in uni-resolution, while we will demonstrate in this work that forecasting tasks on multiple resolutions can benefit long-term prediction performance.

Physics-Informed Modeling of Spatiotemporal Data

As the underlying processes of spatiotemporal data are usually governed by physics laws, physics-informed methods have potential to further improve the performance of neural network models via incorporating inductive bias.

Domain-Specific Knowledge Domain-specific knowledge provides effective inductive bias for solving problems within one specific domain. For example, [Wang *et al.*, 2020; Karpatne *et al.*, 2017] incorporate domain knowledge as regularizations in deep learning models to improve the performance of turbulence simulation and lake temperature prediction respectively. However, when we need a model addressing spatiotemporal modeling tasks from multiple domains, general knowledge of dynamic systems needs to be integrated.

Numerical Methods of Solving Ordinary/Partial Differential Equations (ODEs/PDEs) As the physics laws governing various spatiotemporal processes can be described with ODEs and PDEs in similar forms, incorporating numerical methods generally applicable to ODEs and PDEs can benefit tasks in multiple domains. [Chen *et al.*, 2018; Rubanova *et al.*, 2019] propose scalable back propagation methods through numerical ODE solvers and enable the modeling of irregular time series. [Guen and Thome, 2020; Jiang *et al.*, 2019; Seo *et al.*, 2020] perform convolutions constrained by numerical PDE solvers and achieve better performance compared to unconstrained convolutions.

Koopman Theory Based Methods Koopman theory is based on the insight that the state space of a non-linear dynamic system can be encoded into an infinite-dimensional space where the dynamics is linear [Koopman, 1931]. In practice, people assume the infinite-dimensional space can be approximated with a finite-dimensional space. The key problem is then to find a proper pair of encoder/decoder to map from/to the state space to/from the hidden space.

Traditionally, people construct the encoder/decoder with hand-crafted functions, such as the identity function in Dynamic Mode Decomposition (DMD) [Schmid, 2010], nonlinear functions in Extended DMD (EDMD) [Williams *et al.*, 2015], and kernel functions in Kernel DMD (KDMD) [Kevrekidis *et al.*, 2016]. However, hand-crafted functions may fail to fit complex dynamic systems and are hard to design without domain-specific knowledge. Thus, recent works [Li *et al.*, 2020; Azencot *et al.*, 2020; Lusch *et al.*, 2018] construct encoders/decoders using neural networks as trainable universal approximators. They demonstrate that the combination of neural networks and Koopman theory achieves comparable or even higher performance than the Koopman approximators with hand-crafted mapping functions, while enjoying the ability to generalize to multiple datasets with the same design. [Li *et al.*, 2020] further shows that the integration of Koopman theory allows the model to adapt to new systems with unknown dynamics faster than pure neural networks.

Datasets

Here we introduce the method to construct multi-resolution datasets used in our experiments from raw data in a single resolution. We have attached the constructed datasets to the supplementary material.

YellowCab & GreenCab We construct the YellowCab and GreenCab datasets with the NYCTaxi Trip Record Data[1] between the year 2017 and 2019, which contains the yellow and green taxi trip records including fields of pick-up/drop-off times and locations. The pick-up/drop-off times are precise, while the locations are discrete values from a pre-defined set of regions[2].

To construct one dataset, we first divide the whole time range (3 years) into time windows of equal size, then for each time window and each region, we count the total number of trips starting/ending within the time window and the region as the pick-up/drop-off numbers. The taxi demand forecasting task is to predict the pick-up and drop-off numbers of each time window and each region into the future. We construct two datasets with the yellow and green taxi trip records respectively as YellowCab and GreenCab. Following the practice in [Yao *et al.*, 2019], we choose a 30-minute time window as the finest temporal resolution.

We construct the spatial graph among regions as follows: an undirected edge (i, j) exists iff the average number of trips between the i-th region and the j-th region is at least 1 in 6 hours. The graph contains 1607/374 undirected edges among regions for YellowCab/GreenCab respectively.

We choose [6-hour, 1-day] as additional coarser temporal resolutions, and the New York City United Hospital Fund Neighborhoods (UHF42) as an additional coarser spatial resolution. Figure 1a and Figure 1b visualize the two levels of spatial resolutions of YellowCab and GreenCab.

Solar Energy We construct the Solar Energy dataset with National Renewable Energy Laboratory (NREL)'s solar photovoltaic (PV) power plant data points for the state of Alabama in the United States representing the year 2006[3]. The dataset contains 5-minute power output of 137 solar power plants in Alabama. We follow the practice in [Lai *et al.*, 2018] and select the 10-minute time window as the finest temporal resolution: the whole time range is divided into 10-minute time windows and the value of each time window is the summation of values falling into it from the original data.

We build the spatial graph's adjacency matrix among regions (power plants) using the Gaussian kernel with a threshold: $W_{i,j} = d_{i,j}$ if $d_{i,j} >= \kappa$ else 0, where $d_{i,j} = \exp\left(-\frac{\mathrm{dist}(v_i,v_j)^2}{\sigma^2}\right)$, $\mathrm{dist}(v_i, v_j)$ is the straight line distance from plant v_i to plant v_j, σ is the standard deviation of distances and κ is the threshold. We set $\kappa = 0.95$ for the Solar Energy dataset. The constructed graph has 967 directed edges (including self loops).

We choose [1-hour, 6-hour] as additional coarser temporal resolutions, and we cluster all plants into 10 groups as the additional coarser spatial resolution using KMeans. Figure 1c shows the spatial resolutions of Solar Energy.

[1]https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[2]https://s3.amazonaws.com/nyc-tlc/misc/taxi+_zone_lookup.csv

[3]https://www.nrel.gov/grid/assets/downloads/al-pv-2006.zip

Experimental Details

Baselines

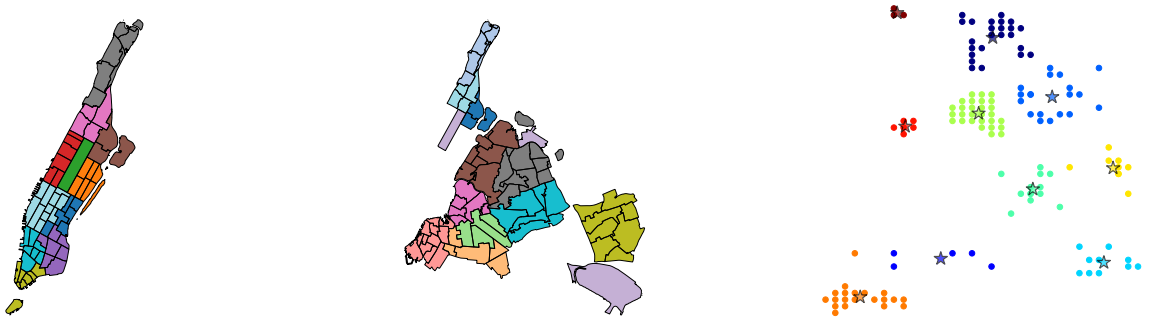
We compare our model ST-KMRN with the following baselines: (1) Historical Averaging (**HA**): We use the averaged value of historical frames with the 1-week period as the prediction. (2) **Static**: We use the value from the last available frame in the input sequence with the 1-week period as the prediction. (3) Gated Recurrent Unit (**GRU**) [Chung *et al.*, 2014] A sequence-to-sequence model with GRU as the architecture for its encoder and decoder. (4) **Informer** [Zhou *et al.*, 2021] : An efficient Transformer-based model with sparse attention mechanism. , which shows state-of-the-art results on long-sequence time series forecasting tasks but does not explicitly involve spatial correlations. (5) **Graph WaveNet** [Wu *et al.*, 2019] : A spatiotemporal forecasting model combining temporal convolutional networks and graph convolutional networks. , which demonstrates compelling results on short-term forecasting tasks. (6) **MTGNN** [Wu *et al.*, 2020] : An improved version of Graph WaveNet, where the graph structure is inferred from data as well as constructed with prior knowledge. (7) **KoopmanAE** [Azencot *et al.*, 2020] : A Koopman theory-informed method for modeling multivariate time series.

As all baseline methods are originally designed for processing data with a single resolution, we preprocess the data as follows: First, we expand data in coarser temporal resolutions to the finest resolution by repeating each element along the temporal dimension. Then we concatenate data in all temporal resolutions to a multivariate time series as the input/label for training and evaluating baselines.

We have tried applying other recent short-term spatiotemporal forecasting works such as AGCRN, GMAN, and DGCRN on our tasks. However, since all the three works are developed and evaluated for short sequence forecasting , their methods and implementation are not optimized for long sequential data: AGCRN and DGCRN adopts conv-lstm-like architectures and GMAN utilize attention among time steps, inducing high computation and memory complexity on long sequential data. AGCRN's and GMAN's memory requirement exceeds the maximum memory size of our GPUs (RTX 2080Ti, 12GiB) even when trained with only 1 batch, while DGCRN takes more than 1 hour to finish training on 1 epoch, prohibiting us from retrieving results with reasonable hardware and amounts of time.

We provide codes and commands for running all baselines in the `README.md` file inside the code folder. Codes of baselines are from the following repositories provided by authors, and we made necessary modifications to enable baselines to load multi-resolution datasets.

- **GRU & Informer**: https://github.com/zhouhaoyi/Informer2020/tree/a732c5dbe7. We use the stable version released at the starting time of this project.
- **Graph WaveNet**: https://github.com/nnzhan/Graph-WaveNet.
- **MTGNN**: https://github.com/nnzhan/MTGNN.



(a) YellowCab (b) GreenCab (c) Solar Energy

Figure 1: Visualization of spatial resolutions. In Figure 1a and Figure 1b, each block bounded with black is one taxi zone in the finer spatial resolution while blocks with the same color is one UHF42 region in the coarser spatial resolution. The geometric centers of blocks are used to build graphs. In Figure 1c, each point is the location of one power plant in the finer spatial resolution, and points with the same color is one cluster in the coarser spatial resolution. Stars are clustering centers.

Training Settings

Datasets and Tasks Table 1 shows statistics and task settings of each dataset. We use sliding windows with strides to generate input/output sequence pairs ordered by starting time. Each pair is one data sample. Then we divide all samples into train/validation/test sets with the ratio 60%/20%/20%. Table 2 shows details of data splits of each dataset.

	YellowCab	GreenCab	Solar Energy
time step size	30min	30min	10min
# time steps	52560	52560	52560
temporal resolutions	[30min, 6h, 1d]	[30min, 6h, 1d]	[10min, 1h, 6h]
# spatial regions	67	76	137
# agg regions	11	12	10
input seq lengths	[1440, 120, 30]	[1440, 120, 30]	[1440, 240, 40]
target temporal resolution	30min	30min	10min
output seq length	480	480	432
max output horizon	10 days	10 days	3 days
target spatial resolution	spatial regions	spatial regions	agg regions

Table 1: Datasets and tasks.

	YellowCab	GreenCab	Solar Energy
time step size	30min	30min	10min
# stride	48 (24 hours)	48 (24 hours)	36 (6 hours)
# train samples	618	618	825
# valid samples	210	210	281
# test samples	210	210	281

Table 2: Details of data splits.

Hyperparameters We provide commands for reproducing results of all baselines and ST-KMRN in the `README.md` file in our code folder, including hyperparameters used in all experiments. Table 3 summarizes some important hyperparameters used in experiments. For all models, we finetune learning rates in {1e-4, 1e-3, 1e-2} based on their performance on the validation data.

	YellowCab	GreenCab	Solar Energy
batch size	2	2	2
epochs	150	150	150
learning rate (Koopman module)	1e-4	1e-4	1e-4
weight decay (Koopman module)	0	0	0
learning rate (all parts except Koopman module)	1e-3	1e-3	1e-3
weight decay (all parts except Koopman module)	1e-4	1e-4	1e-4
loss function	L1 loss	L2 loss	L2 loss

Table 3: Hyperparameters of ST-KMRN.

Computing Resources and Costs Each experiment is conducted with one RTX 2080 Ti GPU with 12GB memory. Table 4 shows the running time of one round experiment of ST-KMRN.

	YellowCab	GreenCab	Solar Energy
running time	3h 52min	4h 32min	5h 33min

Table 4: Running time of ST-KMRN.

Time Complexity and Scalability Time complexity of ST-KMRN is $\mathcal{O}(kL_{in}|V|^2 + k^2 + kL_{out})$, where k is the number of temporal resolutions, $|V|$ is the total number of nodes in the spatial hierarchical graph, L_{in} and L_{out} are the lengths of input and output sequences. On the YellowCab dataset, training 1 epoch takes 357/458/188/69/52/172s for GRU/Informer/Graph WaveNet/MTGNN/KoopmanAE/ST-KMRN respectively. For large-scale datasets with increased k and $|V|$, we can integrate techniques such as sparse attention and neighbor sampling.

Additional Experimental Results

Long-Sequence Forecasting Results With Fully Observed Input Within Multiple Horizons

Table 5 demonstrates results of forecasting from fully observed data within multiple horizons. On all of 3 datasets, ST-KMRN outperforms baselines in most horizons, demonstrating the advantage of ST-KMRN in long-sequence forecasting with multi-resolution data.

Long-Sequence Forecasting Results With Partially Observed Input on GreenCab and Solar Energy

Here we provide the results of forecasting with missing data in input sequences on the GreenCab (Table 6a) and Solar Energy (Table 6b) datasets. Note that we here simulate the practical scenario where data of the highest temporal resolution and the highest spatial resolution suffers from missing while data of other resolutions are fully observed. Since the forecasting task from Solar Energy focuses on the lower spatial resolution, its input from the same resolution is still fully observed and thus the results of HA and Static do not vary with the observation ratios. Similar to results on YellowCab, the results on GreenCab and Solar Energy demonstrate the advantage of ST-KMRN with partially observed input.

Effect of the number of resolutions on the performance.

We evaluate the effect of the number of temporal resolutions on the prediction performance with the YellowCab dataset. Results are shown in Table 7. We find that when the number of available temporal resolutions is reduced gradually from the most coarse resolution, the prediction errors (MAE/RMSE) increase by 12.47%/6.98% (for 1 resolution) and 5.95%/3.27% (for 2 resolutions) respectively for the prediction into future 30 minutes. However, prediction performance for longer terms (6h/1d/10d) does not differ significantly. Results demonstrate that the increase of the number of resolutions in observations will benefit short-term prediction performance.

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.

[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.*, 2018] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6572–6583, 2018.

[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.

[Guen and Thome, 2020] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020.

[Huang *et al.*, 2020] Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. *Advances in Neural Information Processing Systems*, 33, 2020.

[Iakovlev *et al.*, 2020] Valerii Iakovlev, Markus Heinonen, and Harri Lähdesmäki. Learning continuous-time pdes from sparse data with graph neural networks. In *International Conference on Learning Representations*, 2020.

[Jiang *et al.*, 2019] Chiyu Max Jiang, Jingwei Huang, Karthik Kashinath, Philip Marcus, Matthias Niessner, et al. Spherical cnns on unstructured grids. In *International Conference on Learning Representations*, 2019.

[Karpatne *et al.*, 2017] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[Kevrekidis *et al.*, 2016] I Kevrekidis, Clarence W Rowley, and M Williams. A kernel-based method for data-driven koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2016.

[Kipf *et al.*, 2018] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018.

[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.

[Lai *et al.*, 2018] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104, 2018.

[Li *et al.*, 2018] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

[Li *et al.*, 2020] Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional koopman operators for model-based control. In *International Conference on Learning Representations*, 2020.

[Lusch *et al.*, 2018] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.

[Rubanova *et al.*, 2019] Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, pages 5320–5330, 2019.

[Schmid, 2010] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[Seo *et al.*, 2020] Sungyong Seo, Chuizheng Meng, and Yan Liu. Physics-aware difference graph networks for sparsely-observed dynamics. In *International Conference on Learning Representations*, 2020.

[Wang *et al.*, 2020] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1457–1466, 2020.

[Williams *et al.*, 2015] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[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.

[Yan *et al.*, 2018] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[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.

[Yu *et al.*, 2018] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

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

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, page online. AAAI Press, 2021.

Horizon	30min		6h		1d		10d	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	19.43	30.04	10.71	19.89	21.57	34.68	21.93	35.47
Static	*12.50*	*21.87*	**6.95**	*14.78*	*13.45*	*25.00*	*14.00*	*26.32*
GRU	22.69(1.85)	32.98(3.09)	14.75(0.67)	21.99(1.13)	25.57(0.15)	38.34(0.52)	25.62(0.19)	38.21(0.18)
Informer	20.24(1.15)	28.78(1.77)	14.45(0.62)	23.11(0.68)	21.49(1.83)	32.10(2.36)	22.21(2.05)	33.48(2.72)
Graph WaveNet	20.26(0.84)	32.42(0.28)	11.98(0.22)	26.11(0.46)	16.45(0.08)	30.16(0.13)	16.89(0.12)	30.93(0.26)
MTGNN	22.26(1.17)	38.32(1.76)	12.37(0.49)	28.58(0.79)	18.28(0.63)	33.22(0.99)	18.91(0.59)	34.02(0.98)
KoopmanAE	15.45(0.35)	24.11(0.65)	10.31(0.34)	16.92(0.75)	16.71(0.50)	27.20(0.65)	16.92(0.68)	28.11(0.89)
ST-KMRN	**12.27(0.64)**	**18.06(0.47)**	7.53(0.60)	**13.73(0.99)**	**13.32(0.92)**	**23.07(1.32)**	**13.63(0.81)**	**24.09(1.05)**
RelErr	-1.8%	-17.4%	8.3%	-7.1%	-1.0%	-7.7%	-2.6%	-8.5%
RelErrGW	-39.4%	-44.3%	-37.1%	-47.4%	-19.0%	-23.5%	-19.3%	-22.1%

(a) YellowCab

Horizon	30min		6h		1d		10d	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	4.37	6.62	3.25	5.67	3.74	5.67	3.76	5.70
Static	1.95	2.79	1.58	2.29	2.07	3.06	2.08	3.08
GRU	3.54(0.19)	5.17(0.32)	2.33(0.14)	3.52(0.23)	2.67(0.06)	3.89(0.09)	2.67(0.06)	3.89(0.08)
Informer	1.80(0.05)	*2.60(0.05)*	*1.51(0.07)*	*2.18(0.08)*	1.84(0.05)	*2.73(0.07)*	1.93(0.06)	2.82(0.08)
Graph WaveNet	*1.79(0.03)*	2.68(0.03)	1.54(0.04)	2.43(0.05)	*1.80(0.01)*	2.78(0.02)	*1.80(0.01)*	*2.79(0.02)*
MTGNN	2.26(0.02)	3.33(0.01)	1.87(0.01)	2.94(0.02)	2.22(0.02)	3.46(0.01)	2.22(0.02)	3.46(0.01)
KoopmanAE	2.35(0.32)	3.30(0.41)	1.65(0.04)	2.36(0.09)	2.79(0.09)	4.12(0.14)	2.61(0.13)	3.83(0.20)
ST-KMRN	**1.61(0.04)**	**2.37(0.05)**	**1.30(0.03)**	**2.04(0.07)**	**1.65(0.01)**	**2.46(0.01)**	**1.68(0.00)**	**2.48(0.02)**
RelErr	-10.1%	-8.8%	-13.9%	-6.4%	-8.3%	-9.9%	-6.7%	-11.1%
RelErrGW	-10.1%	-11.6%	-15.6%	-16.0%	-8.3%	-11.5%	-6.7%	-11.1%

(b) GreenCab

Horizon	10min		1h		6h		3d	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	*53.1*	*143.3*	*62.5*	147.6	*68.7*	*151.9*	*69.4*	*152.6*
Static	63.1	188.3	63.8	185.7	71.0	179.2	71.8	179.7
GRU	93.2(3.9)	148.0(4.5)	106.1(4.1)	169.4(4.6)	110.1(12.2)	195.6(14.8)	114.7(9.0)	201.6(10.5)
Informer	68.8(4.1)	148.7(6.1)	*62.5(1.2)*	*140.2(5.7)*	79.1(5.0)	161.6(11.0)	81.5(5.2)	171.4(10.1)
Graph WaveNet	124.9(0.2)	310.1(0.2)	123.0(0.3)	301.4(0.3)	136.3(0.1)	291.5(0.3)	135.6(0.1)	290.5(0.2)
MTGNN	117.2(5.8)	296.9(10.5)	115.0(5.7)	288.9(10.2)	129.7(4.8)	282.6(7.3)	130.4(3.5)	283.5(5.7)
KoopmanAE	151.4(3.0)	276.0(0.7)	149.9(1.7)	267.9(0.5)	147.3(1.6)	253.0(0.3)	148.6(2.3)	253.1(0.4)
ST-KMRN	**45.4(1.1)**	**118.9(1.9)**	**44.5(0.1)**	**121.9(1.0)**	**64.3(1.3)**	**140.1(2.0)**	**67.7(1.6)**	**148.3(2.3)**
RelErr	-14.5%	-17.0%	-28.8%	-13.1%	-6.4%	-7.8%	-2.4%	-2.8%
RelErrGW	-63.7%	-61.7%	-63.8%	-59.6%	-52.8%	-51.9%	-50.1%	-49.0%

(c) Solar Energy

Table 5: Forecasting results within multiple horizons 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.

Obs Ratio	0.8		0.6		0.4		0.2	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	3.76	5.70	3.77	5.71	3.77	5.73	3.79	5.78
Static	2.08	3.09	2.14	3.17	2.40	3.47	3.09	4.18
GRU	2.67(0.04)	3.90(0.07)	2.66(0.04)	3.89(0.07)	2.67(0.03)	3.90(0.05)	2.67(0.05)	3.91(0.08)
Informer	1.96(0.07)	2.88(0.10)	2.03(0.07)	2.95(0.08)	2.03(0.02)	2.99(0.07)	2.08(0.05)	3.00(0.07)
Graph WaveNet	*1.77(0.01)*	*2.75(0.00)*	*1.79(0.03)*	*2.78(0.05)*	*1.80(0.02)*	*2.79(0.01)*	*1.82(0.01)*	*2.84(0.02)*
MTGNN	2.09(0.11)	3.24(0.15)	2.21(0.08)	3.46(0.06)	2.21(0.04)	3.43(0.03)	2.23(0.01)	3.50(0.07)
KoopmanAE	2.78(0.55)	4.01(0.72)	2.94(0.44)	4.24(0.50)	2.97(0.45)	4.23(0.50)	2.95(0.50)	4.26(0.59)
ST-KMRN	**1.73(0.02)**	**2.51(0.03)**	**1.72(0.04)**	**2.53(0.03)**	**1.68(0.03)**	**2.47(0.04)**	**1.68(0.04)**	**2.47(0.04)**
RelErr	-2.3%	-8.7%	-3.9%	-9.0%	-6.7%	-11.5%	-7.7%	-13.0%
RelErrGW	-2.3%	-8.7%	-3.9%	-9.0%	-6.7%	-11.5%	-7.7%	-13.0%

(a) GreenCab

Obs Ratio	0.8		0.6		0.4		0.2	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	200.7	255.2	200.7	255.2	200.7	255.2	200.7	255.2
Static	261.1	409.7	261.1	409.7	261.1	409.7	261.1	409.7
GRU	118.3(8.7)	204.2(11.3)	118.5(12.8)	206.0(14.8)	113.9(8.7)	200.4(11.0)	111.8(6.0)	192.1(5.2)
Informer	*91.5(3.0)*	*186.5(7.2)*	90.8(1.0)	182.0(3.8)	*86.7(5.3)*	*177.4(8.1)*	*97.8(7.2)*	*188.8(5.4)*
Graph WaveNet	135.6(0.0)	290.7(0.1)	135.7(0.1)	290.6(0.1)	135.7(0.1)	290.6(0.1)	135.7(0.0)	290.5(0.1)
MTGNN	122.9(8.8)	268.2(16.7)	*79.9(5.3)*	*167.6(19.7)*	106.1(25.3)	225.7(62.3)	107.2(21.4)	233.5(45.2)
KoopmanAE	147.5(0.7)	252.8(0.3)	152.0(7.3)	253.9(1.8)	147.8(0.6)	252.9(0.2)	146.9(1.7)	252.7(0.5)
ST-KMRN	**68.3(1.3)**	**148.0(4.0)**	**67.9(2.1)**	**145.6(4.0)**	**68.1(1.8)**	**149.0(2.8)**	**72.3(2.8)**	**157.7(5.6)**
RelErr	-25.4%	-20.6%	-15.0%	-13.1%	-21.5%	-16.0%	-26.1%	-16.5%
RelErrGW	-49.6%	-49.1%	-50.0%	-49.9%	-49.8%	-48.7%	-46.7%	-45.7%

(b) Solar Energy

Table 6: Forecasting results with partially observed input (Horizon=10d).

	30min		6h		1d		10d	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
1 Res	13.80(0.22)	19.32(0.32)	7.66(0.20)	**12.88(0.18)**	**13.25(0.64)**	**22.68(0.92)**	**13.47(0.64)**	**23.66(0.88)**
	+12.47%	+6.98%	+1.73%	-6.19%	-0.53%	-1.69%	-1.17%	-1.78%
2 Res	13.00(0.87)	18.65(0.44)	8.40(1.52)	13.78(0.97)	13.58(0.36)	22.91(0.17)	13.95(0.51)	23.98(0.34)
	+5.95%	+3.27%	+11.55%	+0.36%	+1.95%	-0.69%	+2.35%	-0.46%
3 Res	**12.27(0.64)**	**18.06(0.47)**	**7.53(0.60)**	13.73(0.99)	13.32(0.92)	23.07(1.32)	13.63(0.81)	24.09(1.05)

Table 7: Effect of the number of resolutions on prediction performance.