Exploring Progress in Multivariate Time Series Forecasting: Comprehensive Benchmarking and Heterogeneity Analysis

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, Gao Cong, Christian S.Jensen, Xueqi Cheng

Abstract—**Multivariate Time Series (MTS) analysis is crucial to understanding and managing complex systems, such as traffic and energy systems, and a variety of approaches to MTS forecasting have been proposed recently. However, we often observe inconsistent or seemingly contradictory performance findings across different studies. This hinders our understanding of the merits of different approaches and slows down progress. We address the need for means of assessing MTS forecasting proposals reliably and fairly, in turn enabling better exploitation of MTS as seen in different applications. Specifically, we first propose BasicTS+, a benchmark designed to enable fair, comprehensive, and reproducible comparison of MTS forecasting solutions. BasicTS+ establishes a unified training pipeline and reasonable settings, enabling an unbiased evaluation. Second, we identify the heterogeneity across different MTS as an important consideration and enable classification of MTS based on their temporal and spatial characteristics. Disregarding this heterogeneity is a prime reason for difficulties in selecting the most promising technical directions. Third, we apply BasicTS+ along with rich datasets to assess the capabilities of more than 45 MTS forecasting solutions. This provides readers with an overall picture of the cutting-edge research on MTS forecasting. The code can be accessed at https://github.com/GestaltCogTeam/BasicTS.**

Index Terms—**benchmarking, multivariate time series, spatial-temporal forecasting, long-term time series forecasting**

I. INTRODUCTION

SENSORS are increasingly being deployed in complex, real-world systems. Readings from such sensors form Multivariate Time Series (MTS) that in turn are used for understanding and operating the host systems. For instance, the PEMS [1] dataset consists of traffic data from critical locations in a transportation system, and the Electricity [2] dataset records the electricity consumption by key clients in a power system. Consequently, MTS forecasting has become fundamental to understanding and operating complex real-world systems, enabling applications such as traffic management [3], emergency management [4], and resource optimization [5].

MTS data analysis must consider both the temporal and spatial aspects of the data [6], [7]. The temporal aspect often encompasses complex dynamics, including non-stationarity, periodicity, and randomness. The spatial aspect concerns interdependencies among time series, known as spatial dependencies [6] or cross-dimension dependencies [8], which can affect prediction accuracy substantially. Effective modeling the complex temporal and spatial aspects of MTS is a key challenge, which also has been addressed in many studies.

Recent MTS forecasting solutions have been based predominantly on deep learning [7], [9], [10], [6], [11], [12]. These solutions often address two prominent and more specific problems, namely Long-term Time Series Forecasting (LTSF) and Spatial-Temporal Forecasting (STF), in which the modeling of temporal and spatial patterns in the data are essential. LTSF solutions are concerned with long-term forecasting and often employ advanced neural networks like Transformers [13] to model long-term temporal dependencies. Notable solutions include efficient Transformers [14], [7], [15], series-level correlations [9], frequency-based solutions [10], and Transformers utilizing patched time series [16], [8]. In contrast, STF solutions aim to improve prediction by effectively modeling spatial correlations. The prevalent approach is to combine Graph Convolution Networks (GCN) [17] with different sequence models [18], [19] to form Spatial-Temporal Graph Neural Networks (STGNN). Examples include combining GCNs with Recurrent Neural Networks (RNN) [6], Convolutional Neural Networks (CNN) [11], and Attention mechanism [20], [21]

While proposals of new solutions include experimental studies, such studies are at times incomparable or seemingly inconsistent. This causes uncertainty on which directions to take and impedes progress towards better solutions. As an example of the current state of affairs, some studies [22], [23], [24], [25] report poor performance of the key baselines DCRNN [6] and GWNet [11], at up to 33% lower than the performance we reproduce. Next, proposals of LTSF solutions [7], [9], [10], [8] usually report evaluations solely using metrics like MAE and MSE based on normalized time series, making prediction errors seem to be very low. An alternative is to perform evaluations on re-normalized data and to report more

Zezhi Shao, Fei Wang, Yongjun Xu, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Xueqi Cheng are with the Institute of Computing Technology, CAS, Beijing 100190, China. Zezhi Shao and Chengqing Yu are also with the University of Chinese Academy of Sciences, Beijing 100049, China. (e-mail: {shaozezhi19b, wangfei, xyj, yuchengqing22b, zhangzhao2021, yaodi, suntao, cxq}@ict.ac.cn)

Wei Wei is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Hubei 430074, China (e-mail: weiw@hust.edu.cn).

Guangyin Jin is with the Tsinghua University, Beijing, China. (e-mail: jinguangyin96@foxmail.com)

Xin Cao is with the School of Computer Science and Engineering, The University of New South Wales, Australia. (e-mail: xin.cao@unsw.edu.au).

Gao Cong is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. (e-mail: gaocong@ntu.edu.sg).

Christian S.Jensen is with the Department of Computer Science, Aalborg University, Denmark. (e-mail: csj@cs.aau.dk).

Corresponding Authours: Fei Wang (wangfei@ict.ac.cn), Yongjun Xu (xyj@ict.ac.cn), and Xueqi Cheng (cxq@ict.ac.cn).

metrics like MAPE and WAPE, which are not affected by the range of data. Issues such as these prevent researchers from judging the strengths and weaknesses of different solutions.

Further, some studies present seemingly contradictory findings in selecting which technical directions to take when pursuing better solutions to LTSF and STF. In relation to the temporal aspect, (i) *the effectiveness of advanced neural networks has been debated* [7], [26], [16], [27]. One study [26] finds that LTSF-Linear, which employs a simple linear layer, significantly outperforms Transformer-based models [7], [9], [10], [15], and the study concludes that Transformer-based architectures are not as effective as previously claimed. However, subsequent studies [16], [28], [27] find that advanced neural networks outperform LTSF-Linear. We find that the difference in model size between these approaches makes it difficult to determine their relative effectiveness. In relation to the spatial aspect, (ii) *the necessity of GCNs has been questioned* [29], [30]. While STGNNs have brought significant improvements, many recent studies highlight the inefficiency of STGNNs and explore alternative means of modeling the dependencies among time series, *e.g.,* normalization [31] [30]. The success of these non-GCN methods indicates the need for a deeper understanding of spatial dependencies and for insight into when these alternative methods are effective.

To mitigate issues such as those exemplified above and to offer insight into the advance achieved, we contribute a comprehensive analysis and comparison of both MTS forecasting datasets and models. First, as we believe that providing a fair, comprehensive, and reproducible benchmark for MTS forecasting can mitigate the current state of affairs and enable progress, we introduce BasicTS+, a benchmark for studying and comparing MTS forecasting solutions. BasicTS+ establishes a unified training pipeline and reasonable evaluation settings. The former resolves inconsistent performance issues caused by unique data and experimental setups in previous studies while the latter enables a more intuitive evaluation of prediction errors. Overall, BasicTS+ facilitates a fair, comprehensive, and reproducible evaluation of over 45 popular MTS forecasting solutions on 20 commonly used datasets.[1]

Second, we address the problem of selecting an appropriate technical approach by studying the impact of the heterogeneity across MTS datasets. We use heterogeneity to refer to completely different patterns observed across different MTS datasets. In the temporal aspect, we classify datasets into those with stable patterns, significant distribution drift, and unclear patterns. In the spatial aspect, we find that spatial sample indistinguishability is a key concept and partition datasets into those with and without significant spatial sample indistinguishability. Experimental studies show that previous conclusions are valid only for certain types of data. For example, basic neural networks [26] only outperform advanced neural networks [7], [9], [10] on datasets without stable temporal patterns, and approaches for modeling spatial dependencies, such as GCN-based approaches, are only effective on datasets

with significant spatial sample indistinguishability. We find that blindly adopting conclusions from previous studies can lead researchers to make misguided inferences.

Moreover, by using BasicTS+ with heterogeneous datasets, we conduct an exhaustive analysis and comparison of popular solutions. Initially, we discuss how to design or select MTS prediction solutions for a given MTS dataset, as well as how to choose suitable datasets for evaluating a given MTS forecasting solution. Subsequently, we present detailed experimental results on the performance and efficiency of popular solutions across comprehensive datasets, shedding light on the advancements made. The objective of these results and discussions is to accelerate progress and facilitate researchers in drawing more reliable conclusions. Additionally, we highlight directions that deserve more attention. In summary, we make the following main contributions:

- We present BasicTS+, the first benchmark specifically designed for fair comparison of MTS forecasting solutions, especially both STF and LTSF solutions. BasicTS+ facilitates evaluation of over 45 popular models on 20 datasets to address the seemly inconsistent performance findings.
- We identify heterogeneity among MTS datasets as a key challenge, and classify datasets based on temporal and spatial characteristics. We find that neglecting heterogeneity is a cause of difficulties in selecting technical directions, and that previous conclusions apply only to certain types of data.
- We conduct an extensive analysis and comparison of popular models using BasicTS+ together with rich heterogeneous datasets. The findings offer valuable insight into the progress already made, aiding researchers in choosing appropriate solutions or datasets, and drawing more reliable conclusions.

The paper is organized as follows. Section II provides discussions of related work on LTSF, STF, and MTS forecasting benchmarking. Section III covers preliminaries and essential definitions. Section IV presents the BasicTS+ benchmark. Section V then delves into the heterogeneity among MTS datasets, and provides hypotheses for explaining seemingly contradictory findings. Section VI reports on the application of BasicTS+ to popular models and provides new insights. Section VII concludes the paper.

II. RELATED WORK

We cover studies related to LTSF and STF, which are the two most prominent topics in recent MTS forecasting studies. We present their goals, techniques, and related open issues. Furthermore, we cover existing MTS benchmarking studies.

A. Long-term Time Series Forecasting

To achieve accurate long-term time series forecasting [32], studies concentrate on capturing the temporal patterns in MTS data, and have proposed methods to efficiently and effectively incorporate longer-term historical information. For example, forecasting future electricity demand over several months or even years in power systems is a typical application scenario,

---

[1]Due to space limitations, not all baselines and datasets are presented in this paper.

where such predictions are crucial for resource optimization and strategic planning.

Early studies typically propose traditional statistical methods (*e.g.,* ARIMA [33] and ETS [34]) or machine learning methods (*e.g.,* GBRT [35] and SVR [36]). These methods often struggle to handle high non-linearity well, and they typically rely heavily on stationarity-related assumptions [3]. With the advent of deep learning [37], [38], studies have embraced more powerful and advanced neural architectures for time series modeling, such as TCN [19], LSTM [39], and Transformer [13]. Among these, Transformer-based models have garnered increasing attention. Informer [7] proposes a Prob-Sparse self-attention mechanism and distilling operation to address the quadratic complexity of the Transformer, leading to significant performance improvements and being recognized as a milestone in LTSF (AAAI 2021 best paper). Subsequently, Autoformer [9] features an efficient auto-correlation mechanism to discover and aggregate information at the series level, while FEDformer [10] proposes an attention mechanism with low-rank approximation in frequency and a mixture of experts to control distribution shifts. Additionally, Pyraformer [15] designs pyramidal attention to effectively describe short and long temporal dependencies with low complexity. Overall, the Transformer architecture is widely regarded as one of the most effective and promising approaches for MTS forecasting.

However, a recent study proposes LTSF-Linear [26] and questions the effectiveness of Transformer architectures. LTSF-Linear employs a simple linear layer and outperforms all the earlier models. It carefully examines every key components of Transformers and concludes that they are ineffective at time series forecasting. This conclusion has subsequently been challenged by studies [16], [27], [28] that employ advanced neural networks to outperform LTSF-Linear. Nevertheless, considering the substantial difference in model size and the small difference in predictive performance, understanding fully the effectiveness of advanced models remains challenging. Furthermore, more exploration is required to understand why a simple linear model can achieve state-of-the-art performance.

B. Spatial-Temporal Forecasting

In contrast to LTSF, spatial-temporal forecasting must contend with not only temporal dynamics in time series but also dependencies among time series. A prime example of this is in traffic management, where predicting future conditions requires data from multiple traffic sensors, clearly highlighting the spatial dependencies among these sensors. Consequently, considerable research has been devoted to effectively capture and model these spatial and temporal patterns.

Early deep learning approaches often employ CNNs to process spatial information and combine CNNs and RNNs [40], [41], [2]. However, as the relationships among time series are usually non-Euclidean, grid-based CNNs may not be optimal for handling spatial dependencies. With the development of GCNs [42], [17], STGNNs [6], [12] have gained increased attention. STGNNs utilize GCNs to model spatial dependencies based on pre-defined prior graphs, and further combine them with sequential models [19], [18], [13]. For example,

models like DCRNN [6], ST-MetaNet [43], and DGCRN [44] incorporate GCNs with RNNs [18] and their variants, and then predict step by step following the seq2seq [39] architecture. Graph WaveNet [11], STGCN [12], and Auto-DSTSGN [45] integrate GCNs with gated TCNs and their variants to facilitate parallel computation. Futhermore, attention mechanisms are used widely in STGNNs, such as GMAN [20], ASTGNN [46]. In addition, neural architecture search solutions [45], [47] have also received widespread attention. However, many recent studies argue that the pre-defined prior graph might be biased, incorrect, or even unavailable in many cases. Thus, they propose to jointly learn the graph structure (*i.e.,* a latent graph) and optimize STGNNs, *e.g.,* AGCRN [48], MTGNN [49], StemGNN [23], GTS [50], DFDGCN [51], and STEP [52].

However, both prior graph-based STGNNs and latent graph-based STGNNs are usually have a complexity ranging from $O(N^2L)$ to $O(N^2L^2)$ due to the graph convolution operation, where N is the number of time series and L is the length of a time series. Consequently, recent studies [53], [54] have questioned the necessity of STGNNs [31], [29], [55], [56] and have explored alternative techniques [30], [31], [57]. For instance, STNorm [31] introduces spatial-temporal normalization, and STID [30] implements a simple yet effective spatial-temporal identity attaching approach. These solutions achieve similar prediction performance as STGNNs but with significantly higher efficiency. The success of these non-GCN solutions highlights the need for a deeper understanding of spatial dependencies and when and how these solutions are effective.

C. MTS Forecasting Benchmarking

Several benchmarking studies have been devoted to MTS forecasting and associated downstream tasks. For example, studies like DGCRN [44], LibCity [58], DL-Traff [59], and our previous work BasicTS [60], use the benchmarks to address STF-based downstream tasks, *e.g.,* urban spatial-temporal forecasting [61]. Similarly, the studies that contribute LTSF-Linear [26] and TimesNet [28] propose benchmarks for LTSF. However, these benchmarks have several limitations. First, they only cover some of the research in either STF or LTSF, and cannot address comprehensively the temporal and spatial aspects of MTS. Second, many of them lack a unified pipeline and instead train each baseline individually with a unique pipeline, which may lead to unfairness. Third, these benchmarks are incapable of covering adequately the issues related to the different technical approaches, to contending with the temporal and spatial aspects of MTS forecasting.

Notably, the motivation and contribution of this study significantly differ from [60]. The focus of this study is to reliably and fairly evaluate MTS forecasting solutions, reveal the heterogeneity across MTS datasets, and address seemingly inconsistent findings in existing studies. This aims to enhance the utilization of MTS in various applications rather than solely proposing benchmarks, surpassing mere software-level contributions. Moreover, even from a software perspective, BasicTS+ has been refactored to adapt and apply to both STF and LTSF tasks (whereas BasicTS [60] is designed only for STF). BasicTS+ also incorporates more extensible features.

TABLE I
INCONSISTENT PERFORMANCE OF GWNET AND DCRNN IN HIGHLY CITED PAPERS. THE PINK BACKGROUND MARKS THE WORST PERFORMANCE, WHILE THE GREEN BACKGROUND MARKS THE PERFORMANCE PRODUCED BY BASICTS+. ASSUMING THAT x AND y ARE THE VALUES REPORTED IN PREVIOUS STUDIES AND BASICTS+, RESPECTIVELY, THEN THE GAP IS DEFINED AS $(x - y)/x \cdot 100\%$.

	Source	PEMS04			PEMS08		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE
GWNet	[22], [62], [23], [63], [57]	25.45	39.70	17.29%	19.13	31.05	12.68%
	[25], [64]	24.89	39.66	17.29%	18.28	30.04	12.15%
	[46]	19.36	31.72	13.31%	15.07	23.85	9.51%
	[24]	28.15	39.88	18.52%	20.30	30.82	13.84%
	BasicTS+	18.80	30.14	13.19%	14.67	23.55	9.46%
	Gap	33.21%↑	24.42%↑	28.78%↑	27.73%↑	23.59%↑	31.64%↑
DCRNN	[22], [62], [23], [63], [57]	24.70	38.12	17.12%	17.86	27.83	11.45%
	[25]	24.63	37.65	17.01%	17.46	27.83	11.39%
	[46]	23.65	37.12	16.05%	18.22	28.29	11.56%
	[24], [48], [64]	21.22	33.44	14.17%	16.82	26.36	10.92%
	BasicTS+	19.66	31.18	13.45%	15.23	24.17	10.21%
	Gap	20.40%↑	12.20%↑	21.43%↑	16.41%↑	14.56%↑	11.67%↑

TABLE II
EVALUATION ON NORMALIZED AND RE-NORMALIZED DATA.

Data	Method	normalized		re-normalized		
		MAE	MSE	MAE	MAPE	WAPE
ETTh1	Autoformer [9]	0.483	0.510	1.74	69.96%	37.61%
	FEDformer [10]	0.460	0.467	1.71	68.92%	36.89%
	Crossformer [8]	0.456	0.461	1.83	64.96%	39.44%
	PatchTST [16]	0.426	0.432	1.60	64.38%	34.49%
ETTh2	Autoformer [9]	0.448	0.433	3.40	59.17%	22.67%
	FEDformer [10]	0.431	0.418	3.35	56.14%	22.33%
	Crossformer [8]	0.453	0.447	3.72	66.76%	24.76%
	PatchTST [16]	0.395	0.390	2.97	55.22%	19.78%

III. PRELIMINARIES

We define key concepts and the forecasting task.

Definition 1: **Multivariate Time Series.** A multivariate time series includes multiple time-dependent variables. It can be expressed as a matrix $\mathbf{X} \in \mathbb{R}^{T \times N}$, where T is the number of time steps and N is the number of variables. We additionally denote the data in time series i ranging from t_1 to t_2 as $\mathbf{X}_{t_1:t_2}^{i}$.

Definition 2: **Multivariate Time Series Forecasting.** Given historical data $\mathbf{X} \in \mathbb{R}^{T_h \times N}$ from the past T_h time steps, multivariate time series forecasting aims to predict $\mathbf{Y} \in \mathbb{R}^{T_f \times N}$ of the T_f nearest future time steps.

IV. BENCHMARK CONSTRUCTION

We present BasicTS+, a benchmark designed for fair, comprehensive, and reproducible evaluation of MTS forecasting solutions, including both STF and LTSF solutions.

A. Unified Training Pipeline

We proceed to delve into the root causes of seemingly inconsistent performance findings and propose in response a unified training pipeline, thereby enabling fair comparison of forecasting models.

1) Inconsistent forecasting performance.: The inconsistencies imply that the forecasting performance of the same solution exhibits notable variations across experimental studies in different papers, even when on the same dataset and with the same experimental settings. To illustrate this, Table I compiles performance findings from studies in a range of papers for two solutions that are often used as baselines: DCRNN [6] and Graph WaveNet [11], on PEMS04 and PEMS08 datasets. All referenced papers employ an identical experimental setup, *i.e.,* they utilize the last 12 time steps to predict the subsequent 12, and they report MAE, RMSE, and MAPE results for the prediction. Each row in the table thus presents performance findings for Graph WaveNet (GWNet in short) or DCRNN as reported in experimental studies in different papers.

We can see a considerable performance variation for each solution across the different papers. We also note that GWNet and DCRNN provide publicly available source code. As such, this variation is likely due to the varying training pipelines employed in the different studies. Furthermore, our benchmark yields markedly improved performance compared to the results reported in the papers, with a maximum gap of 33% (MAE of GWNet on PEMS04). To reduce spurious variations such as those just reported, we conduct a comprehensive analysis of existing codebases, and identify three primary sources of spurious variations: *data processing*, *training configurations*, and *evaluation implementation*. These aspects are often overlooked, although they influence evaluation results substantially.

- *Data Processing:* A crucial step in the learning or inference process involves normalizing raw time series data. Common approaches include min-max normalization and z-score normalization, each exerting varying effects on prediction performance. For example, some studies [46] employ min-max normalization, whereas most studies usually adopt z-score normalization.

- *Training Configurations:* Training configurations include optimization strategies and various training tricks. Different setups have substantial impact on the optimization. For example, most studies [6], [11], [52], [3] employ *masked MAE* for model training, which excludes abnormal values that may affect predictions for normal values adversely. In contrast, some studies [25], [24] adopt *naive MAE* as their optimization function, which tends to yield inferior results. Further, the incorporation of training tricks, such as gradient clipping and curriculum learning, may also influence performance significantly [3].

- *Evaluation Implementation:* While metrics have precise definitions, their implementations can vary across studies, including aspects such as handling outliers, and mini-batch computations [50]. This difference results in significant deviations between testing and actual performance.

2) Implementation of BasicTS+: BasicTS+ introduces a unified training pipeline, as depicted in Figure 1. This mainly incorporates *unified dataloader*, *runner*, and *evaluation* components to address the identified sources of spurious performance variations. The *unified dataloader* is equipped with z-score normalization as the default choice, which generally yields superior performance. Additionally, it adds external temporal features to the raw data such as time-of-day and
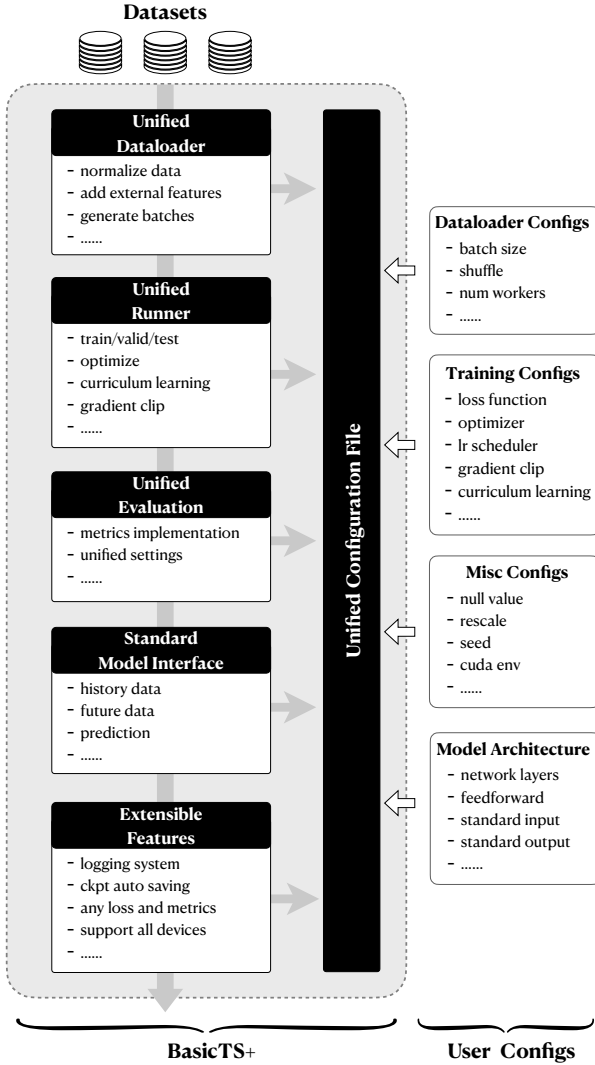
Datasets

Fig. 1. Architecture of BasicTS+.

on *normalized data* (z-score normalized). However, MAE and MSE represent absolute errors that can be influenced significantly by the range of the data, rendering them less intuitive for interpretation. Additionally, evaluating prediction performance on normalized data can yield seemingly very low prediction errors, potentially misleading readers unfamiliar with the details. Thus, some approaches to reporting prediction performance make it difficult for readers to judge whether the prediction performance of the model is satisfactory.

We suggest a practical approach: evaluating on re-normalized data and incorporating additional metrics such as MAPE and WAPE. The performance of important LTSF models on ETTh1 and ETTh2 datasets with normalization and re-normalization are summarized in Table II. We can see that the prediction performance appears less satisfactory on the re-normalized data when considering the high MAPE and WAPE values, in contrast to the seemingly low MAE and MSE values obtained on the normalized data.

In summary, our evaluation is conducted on re-normalized data, employing metrics such as MAE, RMSE, MAPE, and WAPE. Assuming Ω represents the indices of all observed samples, y_i denotes the i-th actual sample, and $\hat{y}_i$ denotes the corresponding prediction, these metrics are defined as follows.

$$\text{MAE}(y,\hat{y}) = \frac{1}{|\Omega|}\sum_{i\in\Omega}|y_i - \hat{y}_i|, \quad \text{RMSE}(y,\hat{y}) = \sqrt{\frac{1}{|\Omega|}\sum_{i\in\Omega}(y_i - \hat{y}_i)^2}$$

$$\text{MAPE}(y,\hat{y}) = \frac{1}{|\Omega|}\sum_{i\in\Omega}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \quad \text{WAPE}(y,\hat{y}) = \frac{\sum_{i\in\Omega}|y_i - \hat{y}_i|}{\sum_{i\in\Omega}|y_i|}$$

(1)

The MAE and RMSE metrics quantify the prediction accuracy, while MAPE and WAPE serve to eliminate the influence of data units. Additionally, for the M4 dataset, we adopt sMAPE, MASE, and OWA. For brevity, we omit their formulations and refer interested readers to the literature [65].

V. HETEROGENEITY ACROSS MTS DATASETS

Next, we put focus on the heterogeneity across MTS datasets and delve into its role in explaining the seemingly contradictory experimental findings that suggest that each of two different technical approaches is the best approach to achieve improved forecasting accuracy. Unlike datasets in computer vision or natural language processing, which often share common patterns, MTS datasets can exhibit very distinct patterns derived from the diverse underlying systems. We thoroughly investigate this heterogeneity and categorize datasets based on characteristics of their temporal and spatial aspects. We argue that different types of patterns entail different solution challenges, implying that specific technical approaches are applicable *only* to particular types of data. Neglecting this data heterogeneity can lead to seemingly conflicting experimental finding and to failure to advocate the right technical approach.

day-of-week attributes. The *unified runner* controls the entire training, validation, and testing procedure. By default, we employ masked MAE as the loss function, which typically outperforms alternatives like naive MAE and MSE. Moreover, *the unified runner* integrates commonly-used training tricks like curriculum learning and gradient clipping. Lastly, the *unified evaluation* component provides standard implementations of metrics including MAE, RMSE, MAPE, WAPE, MSE, and their masked versions. The three components form the foundation that enables BasicTS+ to support fair analyses and comparisons. Given a model that conforms to the *standard model interface*, BasicTS+ can produce evaluation results for that model. Furthermore, BasicTS+ offers many *extensibility features*, such as a logging system, customizable losses and metrics, and compatibility with diverse devices.

B. Evaluation Settings

Evaluation results should be presented in a clear and intuitive manner. In LTSF, many studies adopt metrics such as MAE and MSE and report the prediction performance based

A. Temporal Aspect

We categorize MTS datasets into three types according to their temporal aspect: datasets with clear and stable patterns, datasets with significant distribution drift, and datasets with
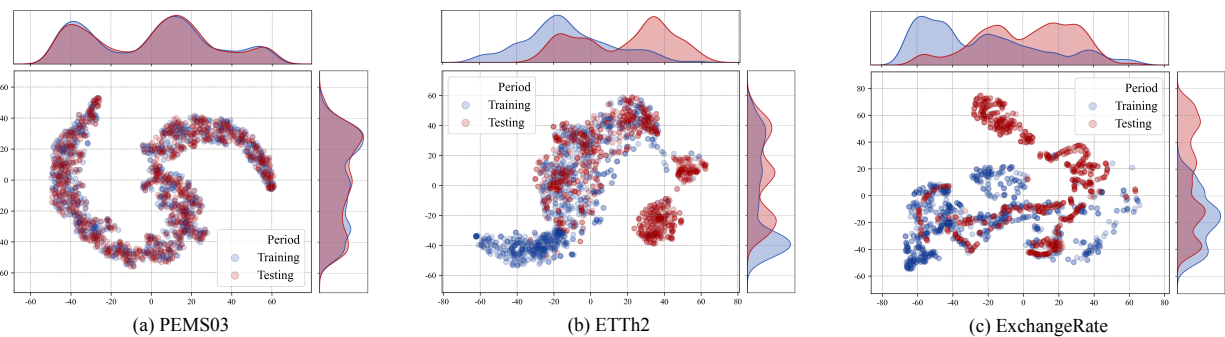
Fig. 2. Visualization of data distribution based on t-SNE and kernel density estimation.
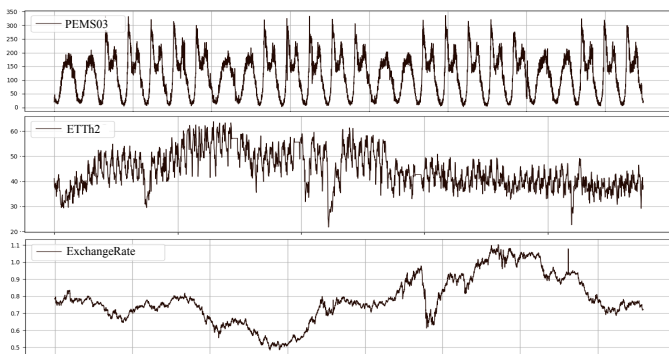

Fig. 3. Distinct temporal patterns in multiple MTS datasets.

unclear patterns. We argue that these types of datasets are progressively less predictable. However, quantifying the predictability [66], [67] remains an unsolved challenge. Thus, we analyze selected datasets through visualizations. Specifically, we chose three typical datasets—PEMS03, ETTh2, and ExchangeRate—and visualized the original time series in Figure 3. To facilitate more intuitive comparisons, we reduce the dimensionality of these datasets to 2D using the t-SNE algorithm [68], and then visualize the data distribution of the training and testing sets with the kernel density estimation algorithm [69], as shown in Figure 2.[2]

We can see significantly distinct patterns across these datasets. First, PEMS03, which records urban traffic flow at different locations, exhibits clear and stable patterns, *i.e.,* periodicity with a fixed period. This pattern conforms to the overall periodicity and stability of urban traffic. Second, ETT contains data from transformer sensors. Although it contains evident cyclic patterns, the period is not fixed, and the mean is shifting, indicating distribution drift. This is because the measured values are affected by external, unobserved factors, such as weather and sensor quality. Third, ExchangeRate records the exchange rates of several currencies and displays minimally discernible patterns. This outcome stems from the

fact that exchange rates are primarily governed by unpredictable factors, such as economic policies. Thus, historical data offers limited value for predictions, particularly for long-term predictions. Additionally, as depicted in Figure 2, the data distributions of the training and testing sets in PEMS03 exhibit a high degree of similarity, whereas in the cases of ETTh2 and ExchangeRate, such similarity is not observed.

Expanding on these insights, we argue that the inherent heterogeneity of MTS data is a key cause of seemingly conflicting findings when comparing advanced neural networks [7], [9], [10], [16] and basic neural networks [26]. Advanced models usually possess strong data fitting capabilities. When coupled with a strong inductive bias, this means that such models imply strong assumptions about data distributions. Conversely, due to their simplicity, basic models like the linear model [26] struggle to capture complex patterns, but also feature relatively weak inductive bias. Considering both the different modeling capabilities of these approaches and the heterogeneous temporal patterns in MTS data we argue that when used on datasets with stable and clear patterns, advanced models should be able to capture complex patterns such as periodicity, while basic linear models remain **under-fitted** due to their limited capacities. In contrast, when used on datasets with significant distribution drift or unclear patterns, advanced models are more likely to capture spurious features present only in the training dataset, thus facing **over-fitting** problems. Based on this discussion, we formulate the following hypothesis:

Hypothesis 1:

1.1 Advanced neural networks outperform basic ones on datasets with clear and stable temporal patterns, while basic neural networks suffer from **under-fitting**.

1.2 Basic neural networks generally outperform advanced ones on datasets with significant distribution drift and datasets with unclear patterns, while advanced neural networks suffer from **over-fitting**.

The study that proposes LTSF-Linear [26] ignores dataset heterogeneity, and conducts experiments on datasets without clear and stable patterns, leading to the biased conclusion that Transformer architectures are ineffective at MTS forecasting. We study this hypothesis experimentally in Section VI-A3.

[2]The results shown in Figure 2 are derived from time series datasets, where samples are obtained by sliding a window of size $P+F$ over the original time series (*i.e.,* the time series in Figure 3). Here, P and F represent the lengths of the historical and future time series, respectively. For the PEMS03 dataset, P and F are set to 12, while for the ETTh2 and ExchangeRate datasets, they are set to 336. The selection of P and F is based on previous works [3], [16], and using different values for P and F yields similar results.
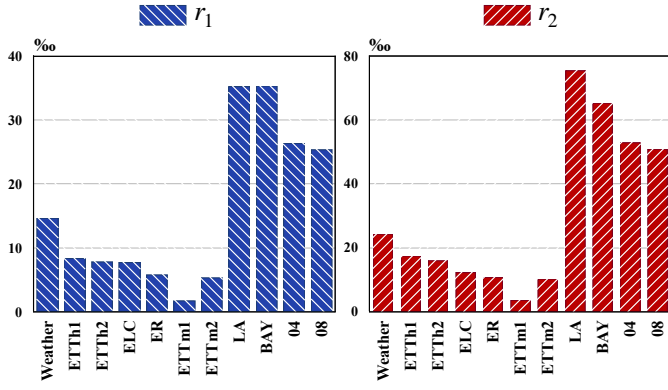
Fig. 4. Spatial indistinguishability in different datasets.

B. Spatial Aspect

Unlike easy-to-see temporal patterns, spatial dependencies are harder to grasp, and it is also more difficult to find clear metrics that allow to distinguish among datasets according to their spatial aspects. Many studies interpret spatial patterns loosely as interactions between time series, and they model them using GCNs, without discussing in depth how to understand and quantify such patterns. Fortunately, two recent studies, ST-Norm [31] and STID [30], point out that the *indistinguishability of samples in the spatial dimension* (spatial indistinguishability in short) gets to the essence of spatial dependencies. In the following, we adopt this idea and, for the first time, design quantitative metrics to distinguish heterogeneous datasets according to their spatial aspect. Specifically, we partition MTS datasets into two types: those with and those without significant spatial indistinguishability, and then we discuss when and how to model spatial dependencies.

In MTS forecasting, samples are generated using a sliding window of size $T_p + T_f$, where T_p and T_f denote the lengths of the historical data and future data. Spatial indistinguishability means that for a given time t, we can expect to generate many samples with similar historical data but different future data. Simple regression models (*e.g.,* using Multi-Layer Perceptions (MLP), RNNs) cannot predict different future data based on similar historical data. Put differently, they cannot distinguish the historical samples [30]. Based on this concept, we propose the following quantitative metrics:

$$r_1 = \frac{\sum_{t,i,j} \mathbb{I}(\mathbf{A}^P_{t,i,j} > e_u \wedge \mathbf{A}^F_{t,i,j} < e_l)}{T \cdot N \cdot N}, \quad r_2 = \frac{\sum_{t,i,j} \mathbb{I}(\mathbf{A}^P_{t,i,j} > e_u \wedge \mathbf{A}^F_{t,i,j} < e_l)}{\sum_{t,i,j} \mathbb{I}(\mathbf{A}^P_{t,i,j} > e_u)}$$

$$\mathbf{A}^P_{t,i,j} = \frac{\mathbf{X}^i_{t-T_p:t} \cdot \mathbf{X}^j_{t-T_p:t}}{\|\mathbf{X}^i_{t-T_p:t}\| \|\mathbf{X}^j_{t-T_p:t}\|}, \quad \mathbf{A}^F_{t,i,j} = \frac{\mathbf{X}^i_{t:t+T_f} \cdot \mathbf{X}^j_{t:t+T_f}}{\|\mathbf{X}^i_{t:t+T_f}\| \|\mathbf{X}^j_{t:t+T_f}\|}$$

(2)

Intuitive Understanding. For a dataset with T time steps and N samples, we construct two similarity matrices, $\mathbf{A}^P, \mathbf{A}^F \in \mathbb{R}^{T \times N \times N}$, representing pairwise similarities among the samples at each time step. Specifically, $\mathbf{A}^*_{t,i,j}$ denotes the similarity between time series i and j at time step t. Using these matrices, we define the total sample count as $T \cdot N \cdot N$, the count of historically similar samples as $\sum_{i,j,t} \mathbb{I}(\mathbf{A}^P_{i,j,t} > e_u)$, and the count of indistinguishable samples as $\sum_{i,j,t} \mathbb{I}(\mathbf{A}^P_{i,j,t} > e_u \wedge \mathbf{A}^F_{i,j,t} < e_l)$. Here, $e_u = 0.9$ and $e_l = 0.5$ are the upper

and lower similarity thresholds, respectively. The indicator function $\mathbb{I}(\cdot)$ returns 1 when the condition is satisfied, and 0 otherwise. We then define two metrics: r_1, the ratio of indistinguishable samples to the total number of samples, and r_2, the ratio of indistinguishable samples to those with similar historical data. These metrics provide complementary insights: r_1 helps determine whether indistinguishability is a major obstacle to improving predictive performance, while r_2 offers a more nuanced evaluation of the degree of indistinguishability.

We calculate the above two metrics for 11 common datasets. The results are shown in Figure 4. We can clearly see that the r_1 and r_2 of ETT, Electricity (ELC), ExchangeRate (ER), and Weather are very low, while the r_1 and r_2 of METR-LA (LA), PEMS-BAY (BAY), PEMS04 (04), and PEMS08 (08) are substantially higher. Interestingly, although these two different groups of datasets have exactly the same format, they are rarely combined in experimental studies. ETT, Electricity, ExchangeRate, and Weather are often used in LTSF studies, where spatial dependencies are not of prime interest. Further, METR-LA, PEMS-BAY, PEMS04, and PEMS08 are used in STF studies, where spatial dependencies take center stage.

Given the above insights, we discuss when and how to model spatial dependencies. First, there is no urgent need to model spatial dependencies on datasets without significant spatial indistinguishability, and forcibly modeling spatial dependencies may even degrade performance. Second, on datasets with significant spatial indistinguishability, modeling spatial dependencies by addressing spatial indistinguishability can improve performance. To be more specific, we discuss how STGNNs [11], [6], ST-Norm [31], and STID [30] work. First, GCNs in STGNNs [11], [6] usually rely on graph structures that conform to the homophily assumption [70], [71], where connected nodes often share similar labels[3]. Therefore, nodes (*i.e.,* time series) with similar historical data but different future data (*i.e.,* labels) are often disconnected. Given such graph structures, GCNs perform message aggregation to make historical data distinguishable. Second, ST-Norm [31] normalizes data on the spatial dimension by separately refining the high-frequency and the local components underlying the input data, making the historical data distinguishable as well. Third, STID [30] proposes a simple yet effective idea of attaching a trainable spatial identity to each time series to distinguish similar historical data. Based on the above discussion, we state the following hypothesis:

> **Hypothesis 2:**
>
> **2.1** On datasets with significant spatial indistinguishability, modeling spatial dependencies by addressing spatial indistinguishability can **improve** performance.
>
> **2.2** On datasets without significant spatial indistinguishability, forcing the modeling towards spatial dependencies may **degrade** performance degradation.

We study this hypothesis in Section VI-C.

[3]In regression, the label is a real-value response corresponding to the instance [72].

VI. EXPERIMENTS

In this section, we conduct extensive experiments to assess our hypotheses and address controversies in technical approaches. In addition, we provide comprehensive analysis and comparison of popular MTS forecasting models based on BasicTS+ and offer insight into the progress already made. Specifically, Section VI-A covers datasets, baselines, and implementation details. Section VI-A3 evaluates the effectiveness of advanced and basic neural networks for LTSF, thus confirming the hypothesis presented in Section V-A V-A. Section VI-C consider when and how to model spatial dependencies, confirming the hypothesis in Section V-B. Section VI-D discusses how to select models or datasets, presents detailed experimental results, and offers insight into the advancements made. All code, datasets, experimental scripts, and results can be accessed through the public GitHub repository at https://github.com/GestaltCogTeam/BasicTS.

A. Experimental Setup

1) Datasets: Following previous LTSF and STF studies [2], [7], [9], [73], [6], we use 14 datasets to conduct experiments, including METR-LA, PEMS-BAY, PEMS03, PEMS04, PEMS07, PEMS08, ETTh1, ETTh2, ETTm1, ETTm2, Electricity, Weather, ExchangeRate, and M4 datasets. Not all the datasets from BasicTS+ are included due to space limitations. The remaining datasets are available via the code repository, including large-scale MTS datasets [74].

2) Baselines: We include popular baselines for which official code is available, including LTSF and STF models. For brevity, we omit their detailed descriptions and simply categorize the baselines based on their technical approaches.

Considering STF models, we cover influential baselines that have high citation counts or offer state-of-the-art performance. First, STGCN [12], DCRNN [6], GWNet [11], DGCRN [44], and D^{2}STGNN [3] are prior-graph-based solutions that rely on pre-defined graphs to indicate spatial dependencies among time series. Second, AGCRN [48], MTGNN [49], StemGNN [23], GTS [50], and STEP [52] are latent-graph-based methods that learn graph structures and optimize STGNNs jointly. Third, we adopt two non-graph based methods, ST-Norm [31] and STID [30].

Considering LTSF models, we cover both advanced and basic neural networks. First, Informer [7], Autoformer [9], FEDformer [10], Triformer [75], Pyraformer [15], Crossformer [8], PatchTST [16] utilize variants of the Transformer to capture long-term historical information. Second, Linear, DLinear, and NLinear utilize a simple linear layer [26].

For a more exhaustive comparison, we also cover three classic time series forecasting models: LGBM [76], DeepAR [77], and NBeats [65]. LGBM is a widely-used gradient boosting framework. DeepAR [77] and NBeats [65] are classic deep learning solutions. These baselines are adopted widely in many industrial applications.

Due to the space limitation, we cannot cover all baselines in BasicTS+; additional baselines are included in the repository, *e.g.,* STGODE [25], NHiTS [78], and TimesNet [28].
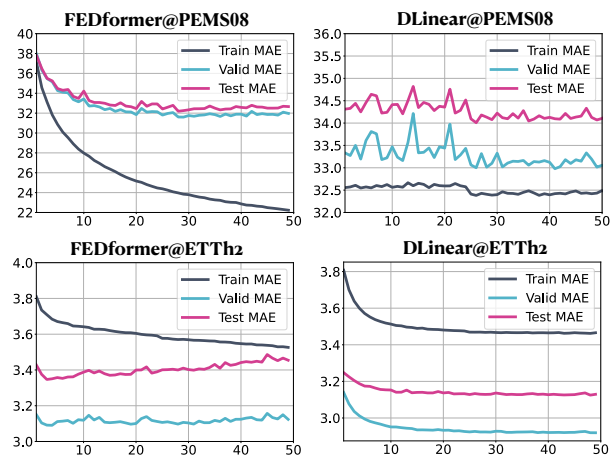

Fig. 5. MAE for varying epochs.

3) Implementation details: For dataset partitioning, we adopt settings consistent with previous work for each dataset. For brevity, we omit the details and refer interested readers to our repository. We set the length of the historical data and future data of the STF task to 12. For the LTSF task, we set the length of future data to 336. We vary the historical length among 96, 192, 336, and 720, and we report the best prediction performance. For error calculations, we report only the *average* error between the forecast time series and the true future time series, due to the space limitation. For the STF task, we employ the MAE, RMSE, MAPE, and WAPE metrics. For the LTSF task, we disregard MAPE, considering that there are many zero values in commonly used LTSF datasets. For the M4 competition dataset, we employ its original settings [79]. In the efficiency studies in Section VI-D, we report the average training time per epoch (in seconds) and the number of model parameter (in million). We set the batch size to 64. If an Out-Of-Memory (OOM) situation occurs, we reduce the batch size by half (to a minimum of 8). All experiments are conducted using a NVIDIA 3090 GPU and 128 GB memory.

4) Hyperparameter tuning: For model implementation, we adopt the public model architecture and hyperparameters. For optimization hyperparameters, such as learning rate and batch size, we also adopt the public settings. Then, we tune these hyperparameters of each model on each dataset via grid search to ensure performance *at least as good as reported in the original paper* (if available). Although using AutoML to tune these hyperparameters may be optimal, we found that manual hyperparameter tuning is acceptable within a certain range. For example, batch sizes of 32, 64, and 128 yield similar performance and do not contradict our findings.

B. Advanced Neural Networks vs. Basic Neural Networks

This subsection studies the performance of advanced models (*e.g.,* Transformers) versus basic models (*e.g.,* linear models) and assesses the hypotheses in Section V-A. We consider four datasets: PEMS04 and PEMS08, which exhibit clear and stable patterns, and ETTh2 and ETTm2, which demonstrate significant distribution drift or unclear patterns. Six baseline models are chosen based on the LTSF-Linear study [26],

TABLE III
PERFORMANCE OF ADVANCED TRANSFORMER MODELS AND BASIC LINEAR MODELS ACROSS HETEROGENEOUS MTS DATASETS.

Methods	PEMS04			PEMS08			ETTh2			ETTm2		
	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE	MAE	RMSE	WAPE
Informer	27.94	44.74	12.84%	26.92	43.79	11.63%	7.12	6.87	47.44%	5.84	7.90	38.97%
Autoformer	34.72	50.33	14.81%	33.75	51.23	14.13%	3.33	**4.91**	22.17%	2.74	4.58	18.27%
FEDformer	**26.89**	**41.46**	12.39%	**25.14**	**39.17**	**10.87%**	**3.27**	4.93	**21.78%**	**2.70**	**4.54**	**17.99%**
Linear	37.42	**62.14**	17.22%	**34.04**	**57.07**	**14.71%**	3.18	5.04	21.19%	2.52	4.24	16.80%
DLinear	**37.51**	62.21	**17.26%**	34.15	57.18	14.76%	**3.13**	**5.00**	20.85%	**2.49**	4.23	16.63%
NLinear	37.62	62.38	17.31%	34.11	57.26	14.74%	3.16	5.06	21.09%	2.49	**4.21**	**16.60%**
Gap	**39.49%↓**	**49.87%↓**	**39.30%↓**	**35.40%↓**	**45.69%↓**	**35.32%↓**	**4.28%↑**	**1.83%↑**	**4.26%↑**	**7.78%↑**	**7.27%↑**	**7.72%↑**

where Informer, Autoformer, and FEDformer are advanced Transformer models, and Linear, DLinear, and NLinear are basic linear models. They all follow the LTSF setup described in Section VI-A3. We report MAE, RMSE, and WAPE. Furthermore, we calculate the performance gap between the best advanced and basic models, as shown in Table III.

First, advanced models generally outperform basic models by a very large margin (green background) on datasets with clear and stable patterns. Second, basic models consistently outperform advanced models on datasets with distribution drifts or unclear patterns. This gap in prediction performance may at first seem puzzling. To intuitively understand why, we visualize the MAE when varying the number of epochs for FEDformer and DLinear on PEMS08 and ETTh2 datasets—see Figure 5. On PEMS08, the training, validation, and testing MAEs of FEDformer start from similar values and keep decreasing. In contrast, DLinear's MAEs, even the training MAE, do not decrease with increasing epochs, which indicates that DLinear suffers from ***under-fitting***. Next, on the ETTh2 dataset, the training MAE of FEDformer keeps decreasing, while its validation and testing MAEs start to increase already when reaching 2 epochs, which indicates that FEDformer suffers from serious ***over-fitting***. These results are consistent with the hypothesis in Section V-A.

We summarize our findings as follows. First, benefiting from their strong modeling capacities, advanced neural networks are far more effective than basic neural networks when the data has clear and stable patterns. Second, models with less inductive bias [80] (*e.g.,* models based on MLPs or a vanilla Transformer [16]) usually perform better when there is no explicit pattern. Moreover, although some recent solutions are positioned as general MTS prediction solutions, we believe that effective general solutions should first perform well on data with clear patterns and should then also consider their performance on time series with less clear patterns.

C. Delving into Spatial Dependencies

Here, we discuss when and how to model spatial dependencies, and we assess the hypothesis in Section V-B. We select four datasets, where METR-LA (LA) and PEMS-BAY (BAY) feature high spatial indistinguishability (see r_1 and r_2 in Section V-B), while ExchangeRate (ER) and ETTm1 feature very low spatial indistinguishability. We choose two baseline models that adopt different approaches to the modeling of the spatial dependencies: STID [30] and AGCRN [48]. STID designs trainable spatial identity embeddings, while AGCRN adopts a GCN-based learning module. Additionally, we remove the spatial modeling components from each of them, obtaining the variant STID* without the spatial identities and the variant AGCRN* with an adjacency matrix set to be the identity matrix.

The results are shown in Table IV. On datasets with significant spatial indistinguishability, the use of both trainable spatial identity embeddings and GCNs can yield significant performance gains. Conversely, on datasets with low spatial indistinguishability, adding these spatial modeling components degrades performance, suggesting that modeling spatial dependencies (or named cross-dimension dependencies) on these datasets is not necessary.

Based on the above discussion, we conclude that spatial indistinguishability is a strong indicator of spatial dependencies and that we do not always need to model spatial dependencies. When there is high spatial indistinguishability in the data, it is purposeful to adopt spatial modeling approaches, *e.g.,* GCNs, normalization [31], and spatial identity [30], to improve performance. In contrast, on datasets with low spatial indistinguishability, designing spatial modeling modules needs to be done with extreme care, as this may cause performance degradation.

D. Performance and Efficiency Benchmarking

So far, we have examined thoroughly the impact of heterogeneity among datasets on the promises of different technical directions and solutions. We find a strong relationship between model architecture and data characteristics. Next, we discuss:

TABLE IV
PERFORMANCE OF STID, AGCRN, AND THEIR VARIANTS ON DATASETS WITH VARYING SPATIAL INDISTINGUISHABILITY.

Data	Metrics	STID	AGCRN	STID*	AGCRN*	Gap
LA	MAE	3.12	3.16	3.58	3.36	**12.85%↑**
	RMSE	6.49	6.44	7.24	6.90	**10.35%↑**
	MAPE	9.13%	8.88%	10.32%	9.66%	**11.53%↑**
BAY	MAE	1.56	1.60	1.80	1.70	**13.33%↑**
	RMSE	3.59	3.67	4.21	3.96	**14.72%↑**
	MAPE	3.53%	3.65%	4.12%	3.92%	**14.32%↑**
ER	MAE	0.0325	0.0455	0.0312	0.0421	**8.07%↓**
	WAPE	4.28%	5.98%	4.11%	5.51%	**8.52%↓**
	MAPE	7.21%	12.03%	6.89%	9.60%	**25.31%↓**
ETTm1	MAE	1.63	2.29	1.41	1.89	**21.16%↓**
	WAPE	35.24%	49.42%	30.64%	40.89%	**20.86%↓**
	MAPE	64.86%	75.64%	55.13%	68.39%	**17.65%↓**

TABLE V
STF ON METR-LA, PEMS-BAY, PEMS03, PEMS04, PEMS07, AND PEMS08 DATASETS.

| Data	Metrics	LGBM	DeepAR	NBeats	STGCN	DCRNN	GWNet	DGCRN	D²STGNN	AGCRN	MTGNN	StemGNN	GTS	STEP	STNorm	STID
METR-LA	MAE	5.03	**3.33**	3.79	3.11	3.03	3.03	2.94	_2.88_	3.16	3.05	3.72	3.13	_2.93_	3.14	**3.12**
	RMSE	9.67	**6.75**	7.74	6.31	6.23	6.12	6.04	_5.91_	6.44	6.16	7.33	6.32	_5.96_	**6.49**	**6.49**
	MAPE	13.12%	**9.76%**	10.69%	8.37%	8.31%	8.17%	_7.79%_	7.83%	8.88%	8.16%	10.43%	8.62%	**8.00%**	**8.84%**	9.13%
	WAPE	8.72%	**5.76%**	6.65%	5.38%	5.24%	5.24%	5.10%	_4.99%_	5.48%	5.28%	6.45%	5.42%	**5.07%**	5.44%	**5.40%**
	Param	-	**0.10**	8.07	0.25	0.37	0.31	**0.20**	0.39	0.75	**0.41**	1.20	38.49	40.48	0.22	**0.12**
	Speed	-	24.48	**11.36**	21.01	94.87	27.70	128.84	152.33	28.22	24.37	**16.19**	52.23	497.26	25.26	_7.50_
PEMS-BAY	MAE	2.10	**1.70**	1.95	1.63	1.59	1.59	1.58	**1.52**	1.60	1.60	1.99	1.68	_1.48_	1.58	**1.56**
	RMSE	4.63	**3.84**	4.96	3.72	3.69	3.68	3.65	**3.53**	3.67	3.71	4.49	3.79	_3.42_	3.65	**3.59**
	MAPE	4.98%	**3.83%**	4.43%	3.69%	3.58%	3.60%	3.52%	**3.44%**	3.65%	3.59%	4.61%	3.78%	_3.31%_	**3.52%**	3.53%
	WAPE	3.37%	**2.72%**	3.13%	2.61%	2.55%	2.55%	2.53%	**2.43%**	2.56%	2.57%	3.19%	2.68%	_2.37%_	2.52%	**2.50%**
	Param	-	**0.11**	8.07	0.31	0.37	0.31	**0.21**	0.40	0.75	**0.57**	1.39	58.67	60.47	0.28	**0.12**
	Speed	-	**42.79**	45.24	**51.91**	223.35	70.24	401.66	374.32	62.07	46.47	**30.30**	179.98	1406.90	63.90	_13.28_
PEMS03	MAE	20.56	**16.63**	19.71	15.83	15.54	_14.59_	14.60	14.63	15.24	**14.85**	16.95	15.41	N/A	**15.32**	15.33
	RMSE	34.19	**28.36**	32.52	27.51	27.18	**25.24**	26.20	26.31	26.65	**25.23**	28.52	26.15	N/A	_25.93_	27.40
	MAPE	22.58%	**17.76%**	19.27%	16.13%	15.62%	15.52%	**14.87%**	15.32%	15.89%	**14.55%**	19.61%	15.39%	N/A	_14.37%_	16.40%
	WAPE	11.82%	**9.56%**	11.34%	9.11%	8.94%	**8.39%**	8.40%	8.42%	8.77%	**8.54%**	9.75%	8.86%	N/A	**8.81%**	8.82%
	Param	-	**0.11**	8.07	0.32	0.37	0.31	**0.21**	0.40	0.75	**0.62**	1.46	25.27	N/A	0.30	**0.12**
	Speed	-	**20.41**	20.60	**25.55**	102.77	30.82	191.13	187.48	28.92	21.66	**13.50**	59.16	N/A	29.58	_5.78_
PEMS04	MAE	26.56	**20.64**	25.30	19.76	19.66	18.80	18.84	_18.32_	19.28	19.13	22.98	21.32	_18.32_	19.21	**18.35**
	RMSE	41.61	**32.35**	39.65	31.51	31.18	30.14	30.48	**29.89**	31.02	31.03	36.00	33.55	_29.91_	32.30	**29.86**
	MAPE	18.96%	**14.28%**	17.66%	13.48%	13.45%	13.19%	12.92%	**12.51%**	13.18%	13.22%	16.56%	14.85%	**12.60%**	13.05%	**12.50%**
	WAPE	12.09%	**9.38%**	11.51%	8.98%	8.94%	8.55%	8.57%	_8.33%_	8.77%	8.70%	10.45%	9.70%	**8.33%**	8.73%	**8.34%**
	Param	-	**0.11**	8.07	0.30	0.37	0.31	**0.21**	0.40	0.75	**0.55**	1.35	16.42	25.47	0.27	**0.12**
	Speed	-	**11.81**	13.01	**14.21**	57.67	16.88	117.09	124.82	16.04	12.01	**7.51**	29.38	745.62	16.87	_3.54_
PEMS07	MAE	29.64	**22.00**	26.14	22.25	21.16	20.44	20.04	_19.49_	**20.68**	21.01	22.50	22.47	N/A	20.59	**19.61**
	RMSE	46.23	**35.44**	42.72	35.83	34.15	33.38	32.86	_32.59_	34.45	**34.14**	36.41	35.42	N/A	34.86	**32.69**
	MAPE	13.52%	**9.31%**	11.37%	9.47%	9.02%	8.71%	8.63%	_8.09%_	**8.77%**	8.92%	9.57%	9.56%	N/A	8.61%	**8.31%**
	WAPE	9.47%	**7.03%**	8.36%	7.11%	6.77%	6.54%	6.41%	_6.23%_	**6.61%**	6.72%	7.19%	7.18%	N/A	6.58%	**6.27%**
	Param	-	**0.12**	8.07	0.59	0.37	0.32	**0.25**	0.41	**0.75**	1.37	3.44	27.18	N/A	0.57	**0.14**
	Speed	-	**39.30**	60.89	**67.14**	324.75	94.51	634.47	748.51	93.79	60.59	**50.93**	512.84	N/A	73.45	_13.63_
PEMS08	MAE	21.29	**16.80**	18.91	16.19	15.23	14.67	14.77	**14.10**	15.78	15.25	16.90	16.92	_14.00_	15.39	**14.21**
	RMSE	33.46	**26.38**	31.39	25.51	24.17	23.55	23.81	**23.36**	24.76	24.22	26.30	26.68	_23.41_	24.80	**23.35**
	MAPE	14.34%	**10.66%**	12.11%	10.82%	10.21%	9.46%	9.77%	**9.33%**	10.42%	10.66%	11.89%	10.88%	_9.50%_	9.91%	**9.32%**
	WAPE	9.15%	**7.24%**	8.14%	6.98%	6.56%	6.32%	6.36%	**6.07%**	6.80%	6.57%	7.28%	7.29%	_6.03%_	6.63%	**6.12%**
	Param	-	**0.10**	8.07	0.23	0.37	0.31	**0.20**	0.39	**0.15**	0.35	1.15	17.25	26.56	0.20	**0.12**
	Speed	-	10.62	**8.01**	**7.46**	35.36	10.34	138.65	109.42	**11.33**	9.57	**6.64**	17.49	420.86	10.98	_3.38_
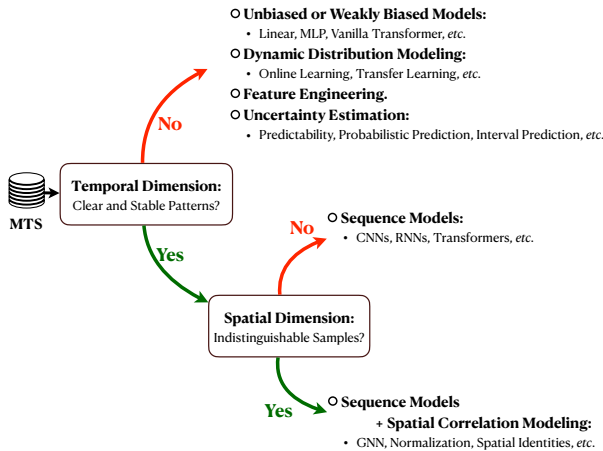
Fig. 6. Road map for selecting or designing MTS models.

(i) how to select or design an MTS forecasting solution for a given dataset and (ii) how to choose datasets suitable for evaluating a given MTS forecasting solution, and we (iii) comprehensively analyze the performance and efficiency of existing solutions using rich datasets and (iv) discuss the progress made and noteworthy research directions.

1) How to select or design MTS solutions for a given dataset.: Patterns in the temporal dimension should be examined first. For data exhibiting significant distribution drift or lacking clear patterns, unbiased or weakly biased models should be chosen, *e.g.,* linear layers, MLPs, or the vanilla Transformer. If the data displays clear and stable patterns, powerful sequence models are a more reasonable option, *e.g.,* TCNs, RNNs, or Transformer architectures. In addition, we need to investigate whether the data has a high sample indistinguishability on the spatial aspect. If so, a spatial dependency modeling module is recommended. Alternative approaches include graph convolution, spatial-temporal normalization, and spatial identity attaching. Moreover, we recommend STID [30] and Linear [26] as baselines. Given their simplicity, we believe that more complex LTSF or STF solutions are only effective if they can significantly outperform these two. We summarize the above discussion in the road map in Figure 6.

2) How to choose suitable datasets for evaluating a given MTS solution.: The key to validating the effectiveness of solutions, which are usually designed to address specific tasks, is to select datasets that align with the task objectives. For instance, STF algorithms often aim to model spatial-temporal dependencies. Thus, datasets with significant spatial dependency are necessary to validate the spatial modeling. LTSF algorithms, on the other hand, aim for generic time series forecasting and should be validated on datasets with and without clear and stable patterns to assess their generalization. However, most LTSF studies only validate on datasets lacking clear patterns like ETT or ExchangeRate. Our experimental results show that this can create an illusion of progress.

Moreover, there are times when our objective is practical, ranking the performance of popular algorithms. In such cases, real compound data is more suitable as it typically encompasses multiple challenges simultaneously. For example, the M4 competition dataset comprises both time series with and

TABLE VI
LTSF ON PEMS04, PEMS08, ETTH1, ETTM1, ELECTRICITY, WEATHER, AND EXCHANGERATE (ER) DATASETS.

Data	Metrics	LGBM	DeepAR	NBeats	Informer	Autoformer	Pyraformer	FEDformer	Triformer	Crossformer	PatchTST	Linear	DLinear	NLinear
PEMS04	MAE	34.55	34.79	**27.95**	27.94	34.72	25.49	26.89	23.81	26.75	25.72	37.42	37.52	37.62
	RMSE	57.74	55.91	46.87	44.74	50.33	41.74	41.46	39.42	45.24	40.13	62.14	62.21	62.38
	WAPE	14.94%	15.83%	12.86%	12.84%	14.81%	11.72%	12.39%	10.95%	12.31%	13.35%	17.22%	17.26%	11.35%
	Param	-	0.11	13.05	12.40	12.07	218.42	18.36	1.69	13.51	2.34	0.11	0.23	0.11
	Speed	-	392.98	31.65	28.62	137.61	25.22	147.75	313.52	338.94	147.42	21.13	21.24	21.27
PEMS08	MAE	38.15	35.58	21.43	26.92	33.75	22.03	25.14	18.74	21.75	19.86	34.04	34.15	34.11
	RMSE	57.74	54.98	38.69	43.79	51.09	38.39	39.17	31.03	36.86	33.44	57.07	57.18	57.26
	WAPE	14.94%	13.14%	9.26%	11.63%	15.37%	9.52%	10.87%	8.13%	9.40%	8.34%	14.71%	14.76%	14.74%
	Param	-	0.10	13.05	11.91	11.37	123.93	17.66	1.64	12.53	2.34	0.11	0.23	0.11
	Speed	-	557.72	18.06	23.34	81.54	20.94	147.89	200.50	224.90	86.81	12.60	12.78	13.04
ETTh1	MAE	1.76	1.94	1.83	2.92	1.74	2.68	1.71	1.80	1.83	1.60	1.60	1.58	1.59
	RMSE	3.34	3.44	3.36	4.60	3.12	4.26	3.15	3.31	3.19	3.08	3.08	3.06	3.09
	WAPE	38.30%	41.89%	39.50%	62.87%	37.61%	57.75%	36.89%	38.72%	39.44%	34.49%	34.47%	34.06%	34.29%
	Param	-	0.10	13.05	11.33	10.54	11.59	16.30	1.59	11.36	0.24	0.11	0.23	0.11
	Speed	-	110.53	3.58	14.18	39.37	11.98	43.44	26.24	11.91	3.90	1.25	1.35	1.28
ETTm1	MAE	1.54	2.21	1.53	2.37	1.93	2.45	1.53	1.57	1.73	1.37	1.39	1.38	1.38
	RMSE	2.92	4.02	2.94	4.20	3.67	3.92	2.89	2.94	3.01	2.78	2.80	2.80	2.81
	WAPE	33.36%	47.72%	33.03%	51.16%	41.69%	52.79%	33.00%	33.83%	37.52%	29.60%	30.02%	29.80%	29.86%
	Param	-	0.10	13.05	11.33	10.54	11.59	16.83	1.59	11.79	0.24	0.11	0.23	0.11
	Speed	-	440.85	8.57	57.32	121.91	46.57	235.17	108.11	98.13	14.77	4.35	4.79	4.40
Electricity	MAE	543.11	283.72	281.63	325.53	295.98	335.23	317.20	275.42	283.86	253.57	256.60	250.08	251.80
	RMSE	8352.53	2998.18	2936.83	2938.56	2933.97	2761.75	2935.53	2968.97	2732.70	2881.53	2896.04	2883.63	2892.13
	WAPE	20.37%	12.27%	10.56%	12.22%	11.11%	12.58%	11.91%	10.33%	10.56%	9.51%	9.62%	9.38%	9.44%
	Param	-	0.11	13.05	12.45	12.14	228.20	17.91	1.69	1.20	0.24	0.11	0.23	0.11
	Speed	-	918.78	55.53	51.00	69.04	70.86	113.94	484.90	198.54	95.92	40.92	39.21	40.71
Weather	MAE	16.66	24.88	12.29	12.88	29.37	38.94	15.61	11.29	11.36	10.85	12.25	12.08	12.02
	RMSE	73.78	73.44	44.57	41.57	84.75	142.50	44.74	40.74	41.27	41.70	43.22	43.35	43.32
	WAPE	9.96%	14.87%	7.29%	7.70%	17.56%	23.28%	9.33%	6.75%	6.79%	6.49%	7.32%	7.22%	7.19%
	Param	-	0.10	13.05	11.38	10.61	21.30	16.90	1.59	11.46	2.21	0.11	0.23	0.11
	Speed	-	439.89	10.77	78.30	99.51	50.52	503.17	120.21	107.23	42.72	8.19	8.73	8.38
ER	MAE	0.0940	0.0608	0.0342	0.0611	0.0366	0.0632	0.0376	0.0367	0.0504	0.0332	0.0352	0.0350	0.0322
	RMSE	0.1531	0.0885	0.0537	0.0803	0.0568	0.0870	0.0578	0.0533	0.0732	0.0525	0.0550	0.0547	0.0508
	WAPE	11.55%	8.00%	4.51%	8.0355%	4.8152%	8.3192%	4.9405%	4.8431%	6.6406%	4.3804%	4.6343%	4.6086%	4.2450%
	Param	-	0.10	13.05	11.33	10.54	12.26	16.31	1.59	0.78	0.24	0.11	0.23	0.11
	Speed	-	64.02	1.39	8.60	13.72	13.03	26.20	15.98	6.65	2.38	0.93	0.95	0.91

without clear and stable patterns. It is important to note that a solution designed specifically for one type of task might not outperform others on such datasets as it contains multiple complex tasks. For instance, SOTA models in STF or LTSF might not yield satisfactory results on the M4 dataset.

3) Experimental Results: First, we present and discuss the detailed performance and efficiency evaluations on LTSF and STF tasks. Then, we select representative solutions from STF and LTSF, along with classic time series solutions, and showcase their results on the complex competition M4 dataset.

The results for LTSF are shown in Table VI. When used on datasets without clear and stable patterns, the state-of-the-art advanced Transformer models [8], [16] and the basic linear models [26] exhibit comparable performance. Considering the simplicity of Linear-based models, *we believe that for LTSF prediction tasks, designing new training strategies or engaging in feature engineering to address distribution drift or ambiguous patterns poses more important challenges than designing increasingly more complex time series forecasting models.* Moreover, on datasets with clear and stable patterns, it is surprising that many recent solutions struggle to outperform the earliest baseline, Informer [7]. Considering that making predictions on such datasets should be a more straightforward task, this raises concerns that the architectures of existing LTSF models might have been over-fitted datasets like ETT, Electricity, Weather, and ExchangeRate that are used commonly in LTSF studies. This reaffirms the importance of selecting appropriate evaluation datasets.

Table V reports the experimental results for STF. Benefiting from the incorporation of prior knowledge, prior-graph-based methods generally perform better than latent-graph-based or non-graph-based methods. Furthermore, it is apparent that learning a graph structure can be very challenging. Among the different solutions, only MTGNN [49] and STEP [52] are capable of learning effective graph structures that do not significantly degrade the prediction performance. *Overall, it is obvious that more intricate network structures yield very limited improvement.* For example, although D^{2}STGNN [3], published in 2022, is the state-of-the-art for STF prediction, its MAE on METR-LA is only 6% higher than that of Graph WaveNet [11], published in 2019. In addition, it is even more surprising that Graph WaveNet [11] and its variant MTGNN are still able to significantly outperform many newer solutions, including StemGNN [23], GTS [50], and others [25], [24], [64]. Therefore, *we find that compared to improving prediction accuracy by designing increasingly complex models, more progress may be achieved by focusing on other important and challenging issues, such as efficiency, graph structure learning.* For example, STID and STNorm are highly efficient and have achieved satisfactory results on most datasets.

In summary, advanced solutions for LTSF and STF represent substantial progress on the modeling of long-term time dependencies and spatial dependencies, respectively. However, complex industrial datasets often contain more complex challenges. Table VII reports the experimental results of representative solutions on the M4 dataset. Specifically, LGBM,

TABLE VII
RESULTS ON THE M4 DATASET.

	Models	LGBM	DeepAR	NBeats	STID	PatchTST
Yearly	sMAPE	14.705	13.886	13.337	13.420	14.158
	MASE	3.565	3.129	3.004	3.071	3.193
	OWA	0.898	0.819	0.786	0.797	0.836
Quarterly	sMAPE	11.358	11.374	9.866	9.869	10.257
	MASE	1.418	1.352	1.136	1.141	1.184
	OWA	1.033	1.009	0.862	0.864	0.898
Monthly	sMAPE	14.559	14.749	12.168	12.624	13.244
	MASE	1.172	1.185	0.897	0.940	1.002
	OWA	1.056	1.068	0.844	0.879	0.93
Others	sMAPE	6.665	6.410	4.635	4.806	4.844
	MASE	4.810	4.769	3.106	3.358	3.166
	OWA	1.460	1.427	0.978	1.035	1.009
Weighted Average	sMAPE	13.430	13.323	11.508	11.755	12.324
	MASE	1.963	1.851	1.550	1.599	1.658
	OWA	1.008	0.975	0.829	0.851	0.888

DeepAR, and NBeats are widely used solutions in industrial applications; STID represents STF prediction solutions, while PatchTST represents LTSF solutions. We follow an existing experimental setup from [28] and report their results on the Yearly, Quarterly, Monthly, and Others subsets, including also their weighted averages. As in the literature [28], we remove the ensemble method in NBeats for fair comparison. Although PatchTST and STID are superior in Tables VI and V, they perform worse than classic algorithms on the M4 dataset.

4) Limitations of Current Studies and Future Directions: There is no doubt that multivariate time series hold significant value in various scientific fields [81], [82], [83]. Although deep learning-based MTS forecasting solutions, particularly in STF and LTSF, have seen substantial advancements, current efforts mainly focus on designing increasingly intricate model architectures. The limitation is that these endeavors appear to be effective only when the data exhibits strong patterns. However, unlike image [84], [85] and natural language data, whose patterns are frequently consistent and stable, time series data can be greatly affected by external factors, resulting in distribution drift or the frequent occurrence of unpredictable changes. Moreover, MTS data in real-world scenarios often face challenges related to insufficient data volume and low data quality [86]. These factors represent key bottlenecks for the broader application of existing research outcomes. Therefore, we emphasize that future research should prioritize more realistic scenarios, such as modeling distribution shifts, predicting with low-quality data, and zero- or few-shot learning.

VII. CONCLUSION

In this study, we address the seemingly inconsistent experimental findings and difficulties in selecting technical directions in the area of Multivariate Time Series (MTS) forecasting, shedding light on the actual advance achieved. First, we introduce a novel benchmark called BasicTS+ that is designed to enable fair and reasonable comparisons of MTS forecasting solutions. By adopting a unified training pipeline, BasicTS+ addresses the issue of inconsistent performance, and provides more reasonable evaluation procedures. Second, we delve into the heterogeneity across MTS datasets. Considering the temporal aspect, we categorize datasets according to whether they exhibit clear and stable patterns, significant distribution drift, or unclear patterns. Considering the spatial aspect, we devise metrics to quantify spatial dependencies and partition datasets into those with and without significant spatial indistinguishability. We emphasize that many conclusions drawn in prior research hold only for certain types of data, and considering these conclusions to be more general can lead researchers to make counterproductive inferences. Additionally, using BasicTS+ and the associated MTS datasets, we conduct an extensive analysis and comparison of popular solutions. These findings offer valuable insight into the progress already made, aiding researchers in choosing appropriate solutions or datasets and drawing more reliable conclusions.

APPENDIX

A. Datasets

- **METR-LA** and **PEMS-BAY** are traffic speed datasets recorded every 5 minutes. The datasets include sensor graphs to indicate spatial dependencies between sensors [6].
- **PEMS03, PEMS04, PEMS07, and PEMS08** are traffic flow datasets recorded every 5 minutes. These datasets include sensor graphs to indicate dependencies between sensors [73].
- **ETTh1, ETTh2, ETTm1, and ETTm2** record temperatures of electricity transformers for use in electric power long-term deployments. The '1'and '2'indicate different transformers, while 'h'and 'm'indicate different sampling every hour and every 15 minutes [7].

TABLE VIII
STATISTICS OF DATASETS.

Dataset	Samples	Variates	Frequency	Time Span	Graph
METR-LA	34272	207	5 mins	4 months	Yes
PEMS-BAY	52116	325	5 mins	6 months	Yes
PEMS03	26208	358	5 mins	3 months	Yes
PEMS04	16992	307	5 mins	2 months	Yes
PEMS07	28224	883	5 mins	4 months	Yes
PEMS08	17856	170	5 mins	2 months	Yes
ETTh1	14400	7	1 hour	20 months	No
ETTh2	14400	7	1 hour	20 months	No
ETTm1	57600	7	15 mins	20 months	No
ETTm2	57600	7	15 mins	20 months	No
Electricity	26304	321	1 hour	2 years	No
Weather	52696	21	10 mins	1 year	No
Exchange Rate	7588	8	1 day	27 Years	No
M4	19-9933	100000	Mixed	N/A	No

- **Electricity** records electricity consumption in kWh by 321 clients every hour from 2012 to 2014 [2].
- **Weather** records 21 meteorological indicators every 10 minutes for the year of 2020 [9].
- **ExchangeRate** collects the daily exchange rates for the currencies of eight countries including Australia, UK, Canada, Switzerland, China, Japan, New Zealand, and Singapore [2].
- **M4** is a well-known competition dataset that contains marketing data collected yearly, quarterly, monthly, weekly, daily, and hourly, with a total of 100,000 time series. These time series are not aligned, which means they have different lengths and start and end times [87].

Table VIII reports summary statistics on the datasets.

B. Baselines

Table IX gives a brief summary of the baselines used in this study.

TABLE IX
OVERVIEW OF BASELINES.

| Methods	Fied	Category	Notes
STGCN	STF	Prior-Graph-based	GCN + TCN
DCRNN			GCN + RNN
GWNet			GCN + TCN
DGCRN			GCN + RNN
D^{2}STGNN			Mixed
AGCRN		Latent-Graph-based	GCN + RNN
MTGNN			GCN + TCN
StemGNN			GCN + TCN + Spectral
GTS			GCN + RNN
STEP			Pretraining-Enhanced
ST-Norm		Non-Graph-based	Normalization
STID			Identity Embedding
Informer	LTSF	Advanced Neural Networks	Efficient Self-Attention
Autoformer			Auto-Correlation
FEDformer			Frequency-Enhanced
Pyraformer			Pyramidal Attention
Triformer			Triangular Attention
Crossformer			Cross-Dimension Attention
PatchTST			Channel Independent + Patchify
Linear		Basic Neural Networks	Vanilla Linear Layer
NLinear			Linear + Normalization
DLinear			Linear + Decomposition
LGBM		Classic	Gradient Boosting
DeepAR			Probabilistic Auto Regressive
NBeats			Neural Basis Expansion

REFERENCES

[1] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, 2001.

[2] G. Lai, W. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *SIGIR*. ACM, 2018.

[3] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, and C. S. Jensen, "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," *Proc. VLDB Endow.*, vol. 15, no. 11, 2022.

[4] H. Wu, H. Zhou, M. Long, and J. Wang, "Interpretable weather forecasting for worldwide stations with a unified deep model," *Nature Machine Intelligence*, 2023.

[5] L. Sun, S. Gong, T. Zhang, F. Jiang, Z. Zhao, J. Chen, and X. Zhang, "SUFS: A generic storage usage forecasting service through adaptive ensemble learning," in *ICDE*. IEEE, 2023.

[6] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, 2018.

[7] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI*, 2021.

[8] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *ICLR*, 2023.

[9] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *NeurIPS*, 2021.

[10] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *ICML*, vol. 162. PMLR, 2022.

[11] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI*. ijcai.org, 2019.

[12] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI*, 2018.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[14] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *NeurIPS*, 2019.

[15] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *ICLR*, 2022.

[16] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *ICLR*, 2023.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[18] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *SSST@EMNLP*. Association for Computational Linguistics, 2014.

[19] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[20] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *AAAI*, 2020.

[21] Z. Shao, F. Wang, Z. Zhang, Y. Fang, G. Jin, and Y. Xu, "Hutformer: Hierarchical u-net transformer for long-term traffic forecasting," *arXiv preprint arXiv:2307.14596*, 2023.

[22] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI*, 2020.

[23] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Spectral temporal graph neural network for multivariate time-series forecasting," in *NeurIPS*, 2020.

[24] Y. Chen, I. Segovia-Dominguez, and Y. R. Gel, "Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting," in *ICML*, vol. 139. PMLR, 2021.

[25] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-temporal graph ODE networks for traffic flow forecasting," in *SIGKDD*. ACM, 2021.

[26] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *AAAI*, 2023.

[27] C. Yu, F. Wang, Z. Shao, T. Sun, L. Wu, and Y. Xu, "Dsformer: A double sampling transformer for multivariate time series long-term prediction," in *CIKM*, 2023.

[28] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *ICLR*, 2023.

[29] X. Liu, Y. Liang, C. Huang, H. Hu, Y. Cao, B. Hooi, and R. Zimmermann, "Do we really need graph neural networks for traffic forecasting?" *CoRR*, vol. abs/2301.12603, 2023.

[30] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *CIKM*. ACM, 2022.

[31] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "St-norm: Spatial and temporal normalization for multi-variate time series forecasting," in *SIGKDD*. ACM, 2021.

[32] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101819, 2023.

[33] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *UKSim 2014.* IEEE, 2014.

[34] E. S. Gardner Jr, "Exponential smoothing: The state of the art," *Journal of forecasting*, vol. 4, no. 1, 1985.

[35] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, 2001.

[36] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *NeurIPS*, 1996.

[37] Y. Xu, X. Liu, X. Cao, C. Huang, E. Liu, S. Qian, X. Liu, Y. Wu, F. Dong, C.-W. Qiu *et al.*, "Artificial intelligence: A powerful paradigm for scientific research," *The Innovation*, vol. 2, no. 4, 2021.

[38] F. Wang, D. Yao, Y. Li, T. Sun, and Z. Zhang, "Ai-enhanced spatial-temporal data-mining technology: New chance for next-generation urban computing," *The Innovation*, vol. 4, no. 2, 2023.

[39] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, 2014.

[40] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *NeurIPS*, 2015.

[41] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *AAAI*, 2018.

[42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016.

[43] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *SIGKDD.* ACM, 2019.

[44] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 1, 2023.

[45] G. Jin, F. Li, J. Zhang, M. Wang, and J. Huang, "Automated dilated spatio-temporal synchronous graph modeling for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, 2023.

[46] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *TKDE*, vol. 34, 2022.

[47] X. Wu, D. Zhang, M. Zhang, C. Guo, B. Yang, and C. S. Jensen, "Autocts+: Joint neural architecture and hyperparameter search for correlated time series forecasting," *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, 2023.

[48] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *NeurIPS*, 2020.

[49] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *SIGKDD.* ACM, 2020.

[50] C. Shang, J. Chen, and J. Bi, "Discrete graph structure learning for forecasting multiple time series," in *ICLR*, 2021.

[51] Y. Li, Z. Shao, Y. Xu, Q. Qiu, Z. Cao, and F. Wang, "Dynamic frequency domain graph convolutional network for traffic forecasting," in *ICASSP.* IEEE, 2024.

[52] Z. Shao, Z. Zhang, F. Wang, and Y. Xu, "Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting," in *SIGKDD.* ACM, 2022.

[53] J. Deng, X. Chen, R. Jiang, D. Yin, Y. Yang, X. Song, and I. W. Tsang, "Disentangling structured components: Towards adaptive, interpretable and scalable time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[54] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, "Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting," in *CIKM.* ACM, 2023.

[55] X. Wang, P. Gu, P. Wang, B. Wang, Z. Zhou, L. Bai, and Y. Wang, "Graph-free learning in graph-structured data: A more efficient and accurate spatiotemporal learning perspective," *CoRR*, vol. abs/2301.11742, 2023.

[56] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "A multi-view multi-task learning framework for multi-variate time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 7665–7680, 2022.

[57] D. Liu, J. Wang, S. Shang, and P. Han, "MSDR: multi-step dependency relation networks for spatial temporal forecasting," in *SIGKDD.* ACM, 2022.

[58] J. Wang, J. Jiang, W. Jiang, C. Li, and W. X. Zhao, "Libcity: An open library for traffic prediction," in *SIGSPATIAL '21: 29th International Conference on Advances in Geographic Information Systems, Virtual Event / Beijing, China, November 2-5, 2021.* ACM, 2021.

[59] R. Jiang, D. Yin, Z. Wang, Y. Wang, J. Deng, H. Liu, Z. Cai, J. Deng, X. Song, and R. Shibasaki, "Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction," in *CIKM.* ACM, 2021.

[60] Y. Liang, Z. Shao, F. Wang, Z. Zhang, T. Sun, and Y. Xu, "Basicts: An open source fair multivariate time series prediction benchmark," in *International Symposium on Benchmarking, Measuring and Optimization.* Springer, 2022.

[61] G. Jin, Y. Liang, Y. Fang, J. Huang, J. Zhang, and Y. Zheng, "Spatio-temporal graph neural networks for predictive learning in urban computing: A survey," *arXiv preprint arXiv:2303.14483*, 2023.

[62] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *AAAI*, vol. 35, no. 5, 2021.

[63] M. Liu, A. Zeng, Z. Xu, Q. Lai, and Q. Xu, "Time series is a special sequence: Forecasting with sample convolution and interaction," *arXiv preprint arXiv:2106.09305*, vol. 1, no. 9, 2021.

[64] J. Choi, H. Choi, J. Hwang, and N. Park, "Graph neural controlled differential equations for traffic forecasting," in *AAAI*, 2022.

[65] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," in *ICLR*, 2019.

[66] P. Xu, L. Yin, Z. Yue, and T. Zhou, "On predictability of time series," *Physica A: Statistical Mechanics and its Applications*, vol. 523, 2019.

[67] J. Garland, R. James, and E. Bradley, "Model-free quantification of time-series predictability," *Physical Review E*, vol. 90, no. 5, 2014.

[68] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[69] Y.-C. Chen, "A tutorial on kernel density estimation and recent advances," *Biostatistics & Epidemiology*, vol. 1, no. 1, 2017.

[70] J. Zhu, R. A. Rossi, A. Rao, T. Mai, N. Lipka, N. K. Ahmed, and D. Koutra, "Graph neural networks with heterophily," in *AAAI*, 2021.

[71] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," in *NeurIPS*, 2020.

[72] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice.* OTexts, 2018.

[73] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *AAAI*, 2019.

[74] X. Liu, Y. Xia, Y. Liang, J. Hu, Y. Wang, L. Bai, C. Huang, Z. Liu, B. Hooi, and R. Zimmermann, "Largest: A benchmark dataset for large-scale traffic forecasting," *NeurIPS*, 2024.

[75] R. Cirstea, C. Guo, B. Yang, T. Kieu, X. Dong, and S. Pan, "Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting," in *IJCAI*, 2022.

[76] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *NeurIPS*, vol. 30, 2017.

[77] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, 2020.

[78] C. Challu, K. G. Olivares, B. N. Oreshkin, F. G. Ramírez, M. M. Canseco, and A. Dubrawski, "NHITS: neural hierarchical interpolation for time series forecasting," in *AAAI*, 2023.

[79] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, 2020.

[80] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[81] T. Zhao, S. Wang, C. Ouyang, M. Chen, C. Liu, J. Zhang, L. Yu, F. Wang, Y. Xie, J. Li *et al.*, "Artificial intelligence for geoscience: Progress, challenges and perspectives," *The Innovation*, 2024.

[82] Y. Xu, F. Wang, Z. An, Q. Wang, and Z. Zhang, "Artificial intelligence for science—bridging data to wisdom," *The Innovation*, vol. 4, no. 6, 2023.

[83] C. Yu, F. Wang, Y. Wang, Z. Shao, T. Sun, D. Yao, and Y. Xu, "Mgsfformer: A multi-granularity spatiotemporal fusion transformer for air quality prediction," *Information Fusion*, 2024.

[84] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang, "Cross-image relational knowledge distillation for semantic segmentation," in *CVPR*, 2022.

[85] L. Huang, Y. Zeng, C. Yang, Z. An, B. Diao, and Y. Xu, "etag: Class-incremental learning via embedding distillation and task-oriented generation," in *AAAI*, 2024.

[86] C. Yu, F. Wang, Z. Shao, T. Qian, Z. Zhang, W. Wei, and Y. Xu, "Ginar: An end-to-end multivariate time series forecasting model suitable for variable missing," *arXiv preprint arXiv:2405.11333*, 2024.

[87] Spyros Makridakis, "M4 dataset," 2018. [Online]. Available: https://github.com/M4Competition/M4-methods/tree/master/Dataset

Zezhi Shao is an Assistant Research Fellow in Institute of Computing Technology, Chinese Academy of Sciences. He received the PhD degree in the Institute of Computing Technology, Chinese Academy of Sciences in 2024. He received the B.E. degree from Shandong University, Jinan, China, in 2019. His research interests include multivariate time series forecasting, spatial-temporal data mining, and graph neural networks. He has published many papers as the first author in top journals and conferences such as TKDE, SIGKDD, VLDB, CIKM, etc.
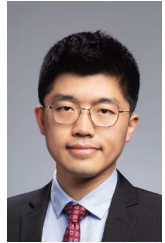
Fei Wang , born in 1988, PhD, associate professor. He received the PhD degree in computer architecture from Institute of Computing Technology, Chinese Academy of Sciences in 2017. From 2017 to 2020, he was a research assistant with the Institute of Technology, Chinese Academy of Sciences. Since 2020, he has been working as associate professor in Institute of Computing Technology, Chinese Academy of Sciences. His main research interest includes spatiotemporal data mining, Information fusion, graph neural networks.

Yongjun Xu is a professor at Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS) in Beijing, China. He received his B.Eng. and Ph.D. degree in computer communication from Xi'an Institute of Posts & Telecoms (China) in 2001 and Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China in 2006, respectively. His current research interests include artificial intelligence systems, and big data processing.

Wei Wei received the PhD degree from the Huazhong University of Science and Technology, Wuhan, China, in 2012. He is currently a professor with the Department of Computer of Science and Technology, Huazhong University of Science and Technology. His current research interests include information retrieval, natural language processing, social computing and recommendation, cross-modal/multimodal computing, deep learning, machine learning and artificial intelligence.

Chengqing Yu received the B.S. degree in Transport Equipment and Control Engineering from Central South University, Changsha, China, in 2019 and M.S. degree in Traffic and Transportation Engineering with Central South University, Changsha, China, in 2022, respectively. He is now PHD student in Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His main research interests include deep learning, reinforcement learning, and time series forecasting.

Zhao Zhang is currently an associate professor at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He received the BE degree in computer science and technology from the Beijing Institute of Technology (BIT), Beijing, China, in 2015, and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2021. His research interests include data mining and applied machine learning, with a special focus on the representation and application of knowledge graphs.

Di Yao is an associate professor in Institute of Computing Technology, Chinese Academy of Sciences (ICT, CAS). He obtained the Ph.D. degree in University of Chinese Academy of Sciences and conducted one year visiting at DMAL, Nanyang Technological University. His research interest lies in spatial temporal data mining, time series analysis, anomaly detection and causal discovery.

Tao Sun is an Assistant Research Fellow in Institute of Computing Technology, Chinese Academy of Sciences. He received the B.S. degree in information and computing science from the Xidian University, Xi'an, China, in 2016. He received the PhD degree in computer architecture from Institute of Computing Technology, Chinese Academy of Sciences in 2022. His research interest falls in the area of spatial-temporal data mining and trajectory data analysis. He has published several papers in journals and conferences, such as CIKM, DASFAA, ICPR, etc.

Guangyin Jin received the PhD degree from the College of Systems Engineering of National University of Defense Technology. His research interest falls in the area of spatial-temporal data mining, graph neural networks and urban computing. So far, he has published more than ten papers in JCR Q1-level international journals such as TKDE, TITS, TIST, TRC, INS, and top international conferences such as AAAI, CIKM, SIGSPATIL.

Xin Cao received the PhD degree from the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2014. He is currently a senior lecturer with the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. His research interests include databases, data mining, and artificial intelligence.
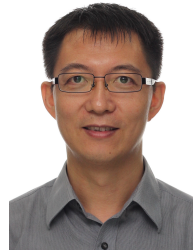
Gao Cong is currently a Professor in the School of Computer Science and Engineering at Nanyang Technological University (NTU). He previously worked at Aalborg University, Denmark, Microsoft Research Asia, and the University of Edinburgh. He received his PhD degree from the National University of Singapore in 2004. His current research interests include spatial data management, ML4DB, spatial-temporal data mining, and recommendation systems.

Christian S. Jensen received the Ph.D. degree from Aalborg University, Denmark, where he is a professor. His research concerns data analytics and management with a focus on temporal and spatiotemporal data. He is a fellow of the ACM and IEEE, and he is a member of the Academia Europaea, the Royal Danish Academy of Sciences and Letters, and the Danish Academy of Technical Sciences.

Xueqi Cheng is a professor in the Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS) and the University of Chinese Academy of Sciences, and the director of the CAS Key Laboratory of Network Data Science and Technology. His main research interests include network science, web search and data mining, big data processing and distributed computing architecture. He has won the Best Paper Award in CIKM (2011), the Best Student Paper Award in SIGIR (2012), and the Best Paper Award Runner up of CIKM (2017).