Extralonger: Toward a Unified Perspective of Spatial-Temporal Factors for Extra-Long-Term Traffic Forecasting

Zhiwei Zhang[1], **Shaojun E**[1],
Fandong Meng[2], **Jie Zhou**[2], **Wenjuan Han**[1]*

[1]Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, China
[2]WeChat AI, Tencent Inc., Beijing, China
`{zhiweizhang, 23140102, wjhan}@bjtu.edu.cn,`
`{fandongmeng, withtomzhou}@tencent.com`

Abstract

Traffic forecasting plays a key role in Intelligent Transportation Systems, and significant strides have been made in this field. However, most existing methods can only predict up to four hours in the future, which doesn't quite meet real-world demands. we identify that the prediction horizon is limited to a few hours mainly due to the separation of temporal and spatial factors, which results in high complexity. Drawing inspiration from Albert Einstein's relativity theory, which suggests space and time are unified and inseparable, we introduce Extralonger, which unifies temporal and spatial factors. Extralonger notably extends the prediction horizon to a week on real-world benchmarks, demonstrating superior efficiency in the training time, inference time, and memory usage. It sets new standards in long-term and extra-long-term scenarios. The code is available at `https://github.com/PlanckChang/Extralonger`.

1 Introduction

Traffic forecasting stands as a pivotal endeavor within Intelligent Transportation Systems (ITS). This task aims to capture the spatial-temporal dynamics in traffic road networks by analyzing historical time steps across monitoring stations and subsequently forecasting future time steps. [1–3] Previous works denote raw traffic data as $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$, where T signifies the time steps length, N represents the count of monitoring stations, and C denotes the raw feature channel. They conceptualize traffic road networks as graphs, with nodes representing monitoring stations and edges between two nodes indicating connectivity. The early works have achieved significant progress in short-term traffic forecasting, whose prediction horizon is within one hour.

In real-world ITS, there exists a considerable demand for prediction horizons that far exceed the typically short-term duration (usually ≤ 1 hour) offered by prevailing deep learning methods. While SSTBAN [4] has initiated exploration into extended forecasting periods of 2 – 4 hours, introducing a long-term traffic forecasting task, we propose an extra-long-term task with the prediction horizons ranging from 0.5 days to 1 week. In this real-world context, the traffic data incurs substantial computational and memory overheads due to the $T \times N$ feature set. Almost all previous approaches to fusing and processing this data exhibit a critical limitation due to treating temporal and spatial dimensions distinctly. Separate processing paradigms for temporal and spatial dimensions necessitate

[*] Corresponding author.

Foundation Models for Science Workshop,38th Conference on Neural Information Processing Systems (NeurIPS 2024).

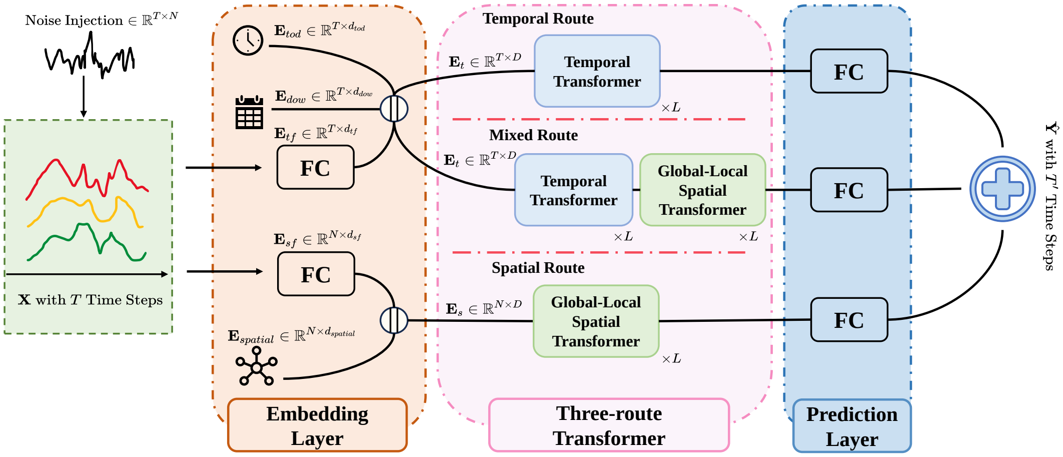

Noise Injection $\in \mathbb{R}^{T\times N}$
$\mathbf{X}$ with T Time Steps
$\mathbf{E}_{tod} \in \mathbb{R}^{T\times d_{tod}}$
$\mathbf{E}_{dow} \in \mathbb{R}^{T\times d_{dow}}$
$\mathbf{E}_{tf} \in \mathbb{R}^{T\times d_{tf}}$
FC
$\mathbf{E}_{sf} \in \mathbb{R}^{N\times d_{sf}}$
FC
$\mathbf{E}_{spatial} \in \mathbb{R}^{N\times d_{spatial}}$
Embedding Layer
Temporal Route
$\mathbf{E}_t \in \mathbb{R}^{T\times D}$
Temporal Transformer
$\times L$
Mixed Route
$\mathbf{E}_t \in \mathbb{R}^{T\times D}$
Temporal Transformer
$\times L$
Global-Local Spatial Transformer
$\times L$
Spatial Route
$\mathbf{E}_s \in \mathbb{R}^{N\times D}$
Global-Local Spatial Transformer
$\times L$
Three-route Transformer
FC
FC
FC
Prediction Layer
$\hat{\mathbf{Y}}$ with T' Time Steps

Figure 1: Overview of the architecture.

repetitive computations across dimensions, leading to inefficiencies. Specifically, temporal feature processing necessitates spatial iterations, while spatial feature processing requires temporal iterations. Consequently, the computational complexity is one order higher than the magnitude of pure time series prediction tasks, e.g., $\mathcal{O}(NT^2 + TN^2)$ in the self-attention mechanism, while memory consumption expands to $\mathcal{O}(TN^2 + NT^2)$, as shown in Table 1. Such computational and memory demands pose significant challenges to extend the prediction horizon into the extra-long-term scenario.

Fortunately, we draw inspiration from Albert Einstein's relativity theory [5] (See Appendix B for more details), on how to make a **unified** perspective on spatial-temporal factors to decrease the computational complexity and memory consumption. We claim that spatial and temporal factors are unified and inseparable, and therefore, they should be treated simultaneously. To address this, we propose the Unified Spatial-Temporal Representation, which circumvents the need to expand dimensionality for embeddings, by incorporating the spatial feature directly into the same time step and the temporal feature into the corresponding node. Using this tailored representation, we design a three-route Transformer-based architecture for traffic forecasting, named Extralonger. Our contributions are threefold:

- We propose Extralonger, founded on Unified Spatial-Temporal Representation. Additionally, we introduce the extra-long-term traffic forecasting task.
- Extralonger achieves a one-order reduction in both computational complexity, $\mathcal{O}(NT^2 + TN^2) \to \mathcal{O}(T^2 + N^2)$, and memory usage, $\mathcal{O}(NT^2 + TN^2) \to \mathcal{O}(T^2 + N^2)$. Extralonger tremendously excels the baselines in terms of memory consumption, training efficiency, and inference speed. Especially, in the longest step scenario, Extralonger achieves a 172× reduction in memory usage, 500× increase in training speed, 385× increase in inference speed than the prior best performance method (Figure 5).
- Extralonger first successfully extends the traffic forecasting horizon from 2-4 hours to 1 week. It outperforms baselines in both long-term scenarios (2-4 hours, as shown in Table 2) and extra-long-term scenarios (0.5 days to 1 week, detailed in Table 3).

2 Preliminary

Given traffic road network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of N nodes (i.e., monitoring stations) and $\mathcal{E}$ is the set of edges. An edge exists if two nodes are connected, from which we derive the adjacency matrix $\mathcal{A}$. The historical traffic signal data are represented as $\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{T-1}] \in \mathbb{R}^{T\times N\times C}$, where T denotes the length of historical time steps, C represents the raw feature, with a value of 1 for a specific feature such as flow, speed or occupation.

The goal of traffic forecasting is to find a model $\mathbb{F}$ to predict the future traffic signal data $\hat{\mathbf{Y}} = [\hat{\mathbf{x}}_T, \hat{\mathbf{x}}_{T+1}, \ldots, \hat{\mathbf{x}}_{T+T'-1}]$, where T' is the length of future time steps, which can be formulated as:

$$\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{T-1}] \xrightarrow{\mathbb{F}(\cdot|\Theta)} \hat{\mathbf{Y}} = [\hat{\mathbf{x}}_T, \hat{\mathbf{x}}_{T+1}, \ldots, \hat{\mathbf{x}}_{T+T'-1}] \tag{1}$$

where Θ is the parameters, conditional on which, the model $\mathbb{F}(\cdot)$ has the minimum loss calculated with the ground truth $\mathbf{Y}$.

3 Extralonger

We first introduce the Unified Spatial-Temporal Representation in Section 3.1, followed by a step-by-step description of the architecture of Extralonger in Section 3.2.

3.1 Unified Spatial-Temporal Representation

3.1.1 Classical Representation

The pipeline with the classical representation $\mathbf{E} \in \mathbb{R}^{T \times N \times D}$ for previous traffic forecasting models is illustrated at the top of Figure 2 and formulated as follows :

$$\mathbf{X} \in \mathbb{R}^{T \times N \times C} \xrightarrow{\text{Linear}(C)} \mathbf{E} \in \mathbb{R}^{T \times N \times D} \xrightarrow{\mathbb{F}(\cdot|\Theta)} \hat{\mathbf{Y}} \in \mathbb{R}^{T' \times N \times C} \tag{2}$$

where D is the feature dimension. All current traffic forecasting methods, to the best of our knowledge, utilize the classical representation. They linearly project the raw data to the $\mathbf{E}$, which serves as the input to the forecasting model $\mathbb{F}$ involving temporal and spatial modules. The combination of the two modules is summarized as temporal-first, spatial-first, and spatial-temporal simultaneous [6], as depicted by the gray bidirectional arrow in Figure 2.

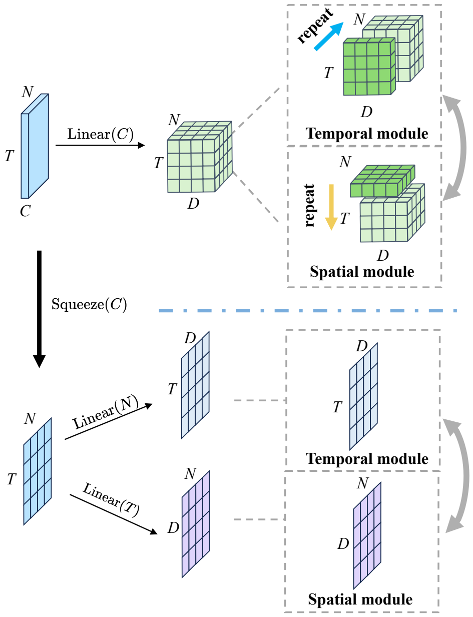


Figure 2: Classical pipeline (TOP) and New pipeline (BOTTOM).

We categorize prior methods into four main groups: RNN-based methods, attention-based methods, Transformer-based methods, and CNN-based methods. The former three groups of methods necessitate spatial iterations when processing temporal features, and require temporal iterations when processing spatial features. Essentially, while processing features for one dimension, the other dimension acts as a role of batch size. Consequently, the computational complexity of RNN-based methods is $\mathcal{O}(TN)$. Considering the diverse variants of attention-based and Transformer-based methods, we only give the complexity of the representative self-attention mechanism, i.e., $\mathcal{O}(NT^2 + TN^2)$.

Memory usage analysis is complex and depends on many factors beyond the scope of our paper, including the parallelization framework, optimizer type, and decisions regarding intermediate state maintenance [7]. Ignoring the constant memory footprint of model parameters and optimizer state, we approximate memory usage as $\mathcal{O}(TN)$ for RNN-based methods and $\mathcal{O}(NT^2 + TN^2)$ for attention-based and Transformer-based methods.

Especially, CNN-based methods [8, 9] can treat the spatial and temporal dimensions jointly. Similar to image data $\mathbb{R}^{Height \times Width \times Channel}$ in computer vision, CNN-based methods correspondingly process the traffic data $\mathbb{R}^{T \times N \times D}$. Therefore, the computational complexity of CNN-based methods follows the complexity of image processing, resulting in $\mathcal{O}(k^2TN)$, where k denotes the kernel size. Since CNN-based methods leverage GNNs as the spatial module, the total complexity is $\mathcal{O}(k^2TN + TN)$ for time complexity and $\mathcal{O}(TN)$ for memory usage.

These limitations of time and space complexity restrict the ability of existing methods to extend the prediction horizon to a weekly level. More details are given in Appendix C and Table 1.

Table 1: Comparison of time complexity, space complexity and maximum path lengths of mainstream methods. T is the time steps length, N is the node number and k is the kernel size of CNN. The maximum path length of the self-attention mechanism with classical representation scales as $\mathcal{O}(2)$ due to independent computations along the temporal and spatial dimensions.

Method	Time Complexity	Space Complexity	Maximum Path Length
RNN-based (Classical)	$\mathcal{O}(TN)$	$\mathcal{O}(TN)$	$\mathcal{O}(T+N)$
CNN-based (Classical)	$\mathcal{O}(k^2TN+TN)$	$\mathcal{O}(TN)$	$\mathcal{O}(log_kT+log_kN)$
Self-Attention (Classical)	$\mathcal{O}(NT^2+TN^2)$	$\mathcal{O}(NT^2+TN^2)$	$\mathcal{O}(2)$
Self-Attention (Unified)	$\mathcal{O}(T^2+N^2)$	$\mathcal{O}(T^2+N^2)$	$\mathcal{O}(1)$

3.1.2 New Representation

To address high-complexity issues, we rethink the classical representation and propose the Unified Spatial-Temporal Representation. Since feature number C equals 1, we squeeze it directly. We adopt two fully connected linear layers and combine them with other prior embeddings (Section 3.2.1) to obtain the temporal and spatial representations: $\mathbf{E}_t$ and $\mathbf{E}_s$. With these two representations, the new pipeline is shown at the bottom of Figure 2 and formulated as follows:

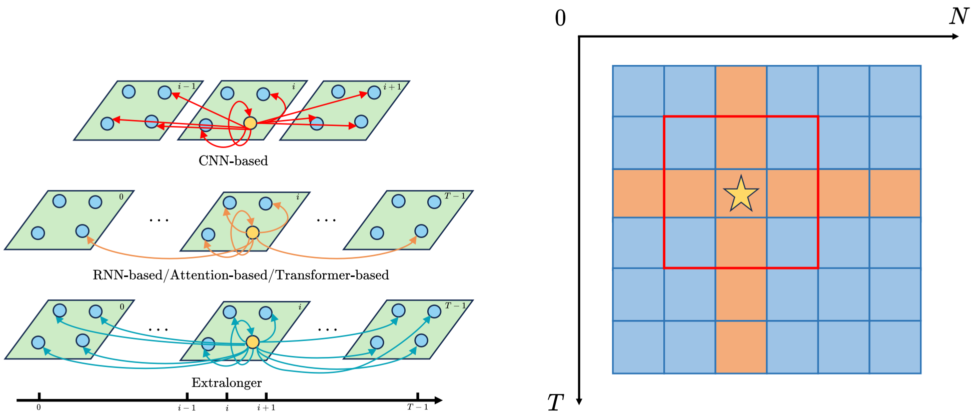
$$\mathbf{X} \in \mathbb{R}^{T\times N\times C} \xrightarrow{\text{Squeeze}(C)} \mathbb{R}^{T\times N} \left\{ \begin{array}{c} \xrightarrow{\text{Linear}(N)} \mathbf{E}_t \in \mathbb{R}^{T\times D} \\ \xrightarrow{\text{Linear}(T)} \mathbf{E}_s \in \mathbb{R}^{N\times D} \end{array} \right\} \xrightarrow{\mathbb{F}(\cdot|\Theta)} \hat{\mathbf{Y}} \in \mathbb{R}^{T'\times N} \tag{3}$$

The new representation integrates the whole nodes to the same time step directly and integrates all the time steps to the same node as the linear layers play a channel-mixed role. Hence, for every i in the range $[0, T-1]$, each element in the set $\{\mathbf{E}_t[i,j] : j \in [0, D-1]\}$ contains information about all the nodes at time step i. Similarly, for every n in the range $[0, N-1]$, each element in the set $\{\mathbf{E}_s[n,m] : m \in [0, D-1]\}$ contains information about all the time steps at node n. This operation reflects the idea of **unification**: each time step incorporates information from all nodes, while each node integrates information across all time steps, which contrasts with the classical representation that relies solely on the feature itself at a particular node and time step. Early methods that treat spatial and temporal dimensions separately require repetitive operations, leading to increased complexity. However, the Unified Spatial-Temporal Representation overcomes this issue by reducing one dimension. Additionally, our novel representation offers the following three advantages.

Complexity Reduction. The Unified Spatial-Temporal Representation inherently reduces computational complexity and memory usage compared to the classical representation. This stems from that one dimension less than the classical representation. This eliminates the need for repetitive operations across the other dimension. Since our model employs a Transformer-based architecture denoted as $\mathbb{F}$, the time complexity decreases from $\mathcal{O}(NT^2+TN^2)$ to $\mathcal{O}(T^2+N^2)$. Notably, if incorporating efficient attention mechanisms [10], the computational complexity could be further reduced to $\mathcal{O}(TlogT+NlogN)$. Similarly, memory usage reduces from $\mathcal{O}(NT^2+TN^2)$ to $\mathcal{O}(T^2+N^2)$.

Simultaneous Aggregation. The traffic signal transmits between the upstream nodes and the downstream nodes with time going by, so the signal is related to different nodes and different times, not just fixing one dimension and varying another one. Extralonger allows each node to be simultaneously linked to all other nodes at the current time step, as well as to the nodes from all historical and future time steps. This representation enables Extralonger to effectively capture the complex spatial-temporal dynamics underlying traffic trends. Conversely, RNN-based, attention-based and Transformer-based methods can only aggregate the nodes at the same time step or the same node among different time steps. Treating traffic data as a grid, CNN-based methods are unable to simultaneously consider all nodes at the same time step. Instead, they can only aggregate the nodes within adjacent time steps due to the inherent receptive field. The aggregation comparison of ours and other methods is shown in Figure 3(a).

Complete Receptive Field. The importance of a complete receptive field from both spatial and temporal dimensions for capturing long-range dependencies in traffic forecasting has been well-established in prior work [6, 9, 11]. Extralonger achieves a complete receptive field compared to existing approaches. This stems from the inherent characteristic of the Unified Spatial-Temporal Representation and the self-attention mechanism employed by Extralonger. As illustrated in the three-route Transformer architecture in Section 3.2, the self-attention mechanism enables comprehensive connectivity among all elements in the sets $\{\mathbf{E}_t[i,:] : i \in [0, T-1]\}$ and $\{\mathbf{E}_s[n,:] : n \in [0, N-1]\}$.



(a) The aggregation comparison. (b) The receptive field comparison.

Figure 3: Given the yellow circle and star are the analysis target at t time step, (a) only Extralonger aggregates along different nodes and different time steps globally; (b) CNN-based methods aggregate the information within the restricted receptive field (red box), and the other early methods aggregate along the temporal and spatial dimensions separately (two orange belts). Capturing temporal and spatial dependency globally with the Unified Spatial-Temporal Representation (whole area), Extralonger does simultaneously and efficiently.

This full connection facilitates the aggregation of information from any node across the entire road network among all time steps, i.e., the complete receptive field, as depicted by all the squares in Figure 3(b). Consequently, the maximum operation path length in Extralonger is $\mathcal{O}(1)$. In contrast, RNN-based, attention-based and Transformer-based methods typically aggregate information along the temporal and spatial dimensions separately, leading to an axial receptive field (orange belts in Figure 3(b)). CNN-based methods are constrained by their inherent limitations on convolution size (red box in Figure 3(b)). While CNN-based methods can operate along both dimensions, they cannot make sure all nodes in the complete road network and all time steps are in the receptive field. Moreover, unlike the image size that is fixed or can be preprocessed to the same size in computer vision tasks, CNN-based models are not robust enough due to the T varying with the prediction horizon.

3.2 Architecture of Extralonger

Founded on the Unified Spatial-Temporal Representation, we propose Extralonger, comprising three parts: embedding layer, three-route Transformer, and prediction layer, as shown in Figure 1. The embedding layer transforms the raw traffic data into an embedding space. The resulting embedding consists of two components, the spatial representation $\mathbf{E}_s$ and the temporal representation $\mathbf{E}_t$. The three-route Transformer is the heart of Extralonger and is comprised of three parallel routes: a temporal route, a spatial route, and a mixed route. Notably, all three routes can simultaneously process information from both spatial and temporal dimensions due to the adoption of the Unified Spatial-Temporal Representation. Each route emphasizes specific aspects of the data. The temporal Transformer in the temporal route and the mixed route is implemented as a standard Transformer encoder, and its detailed description is omitted for brevity. The Global-Local Spatial Transformer is specifically designed for the spatial route and the mixed route. Finally, projected by the prediction layer first, the outputs of the three routes are added up with a hand-crafted weight.

3.2.1 Embedding Layer

We first inject noise into $\mathbf{X}$ with learnable parameters (More details in Appendix E). Then, we use two fully connected linear layers to transform $\mathbf{X}$ to feature embeddings: $\mathbf{E}_{tf} \in \mathbb{R}^{T\times d_{tf}}$ and $\mathbf{E}_{sf} \in \mathbb{R}^{N\times d_{sf}}$. We also utilize learnable embedding to capture the periodicity. Specifically, we employ $\mathbf{E}_{tod} \in \mathbb{R}^{T\times d_{tod}}$ for the 24-hour cycle (namely, timestamp-of-day embedding) and $\mathbf{E}_{dow} \in \mathbb{R}^{T\times d_{dow}}$ for the 7-day cycle (namely, day-of-week embedding). We adopt a learnable spatial embedding $\mathbf{E}_{spatial} \in \mathbb{R}^{N\times d_{spatial}}$, introduced by GWNet [8], to capture the spatial dynamics. The temporal representation $\mathbf{E}_t$ is formed by concatenating the aforementioned embeddings: $\mathbf{E}_t =$

$\mathbf{E}_{tf}||\mathbf{E}_{tod}||\mathbf{E}_{dow} \in \mathbb{R}^{T\times D}$, where $||$ denotes concatenation. The spatial representation $\mathbf{E}_s$ is informed by concatenating embeddings: $\mathbf{E}_s = \mathbf{E}_{sf}||\mathbf{E}_{spatial} \in \mathbb{R}^{N\times D}$.

3.2.2 Global-Local Spatial Transformer

The proposed Global-Local Spatial Transformer is specifically tailored to exploit the inherent topological characteristics of the spatial domain. It integrates the road network's prior information into the self-attention mechanism. By incorporating dynamic dependencies derived from all nodes and focusing on neighboring nodes to aggregate local correlations, the Global-Local Spatial Transformer explicitly leverages both global and local spatial information, as shown in Figure 4. More details about motivation and insights are given in Appendix D.

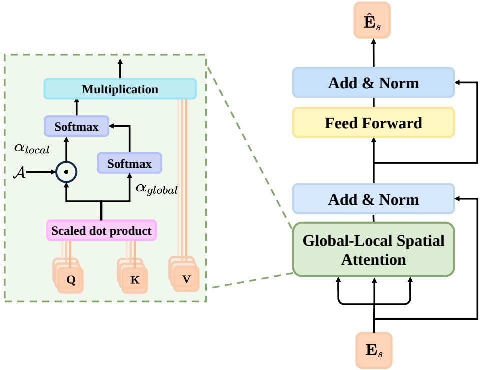


Figure 4: Global-Local Spatial Transformer.

Given $\mathbf{Q} = \mathbf{E}_s\mathbf{W}_Q$, $\mathbf{K} = \mathbf{E}_s\mathbf{W}_K$, $\mathbf{V} = \mathbf{E}_s\mathbf{W}_V$, where $\mathbf{W}_Q$, $\mathbf{W}_K$ and $\mathbf{W}_V$ are learnable parameters, the Global-Attention score before softmax is followed as:

$$\alpha_{global}(\mathbf{Q}, \mathbf{K}) = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}} \tag{4}$$

Then, we inject the road network's prior information by using the adjacency matrix $\mathcal{A}$ as a mask to obtain the Local-Attention score before softmax, formulated as:

$$\alpha_{local} = \alpha_{global} \odot \mathcal{A} \tag{5}$$

where $\odot$ denotes the element-wise product. We designate it as *local* because only information from adjacent nodes is considered, while non-adjacent nodes are masked out. Both α_{local} and α_{global} are subjected to a softmax function to obtain normalized attention weights. Global-Local Spatial Attention (GLSAtt) is obtained by weighted summation with $\mathbf{V}$.

$$\text{GLSAtt} = (\text{Softmax}(\alpha_{local}) + \text{Softmax}(\alpha_{global}))\frac{\mathbf{V}}{2} \tag{6}$$

We sequentially apply layer normalization (LN), skip connections, and a feed-forward network (FFN), the same procedure as the vanilla Transformer, to yield the output $\hat{\mathbf{E}}_s$.

$$\mathbf{Z} = \text{LN}(\text{GLSAtt}(\mathbf{E}_s)) + \mathbf{E}_s \tag{7}$$

$$\hat{\mathbf{E}}_s = \text{LN}(\text{FFN}(\mathbf{Z})) + \mathbf{Z} \tag{8}$$

3.2.3 Prediction Layer

We employ linear layers to project the output of the three-route Transformer to predict the corresponding T' time steps, which is necessary for the asymmetrical $T - T'$ setup. Subsequently, the projected outputs are aggregated using hand-crafted weights, resulting in $\hat{\mathbf{Y}} \in \mathbb{R}^{T'\times N}$. The Huber loss [12] is then calculated between $\hat{\mathbf{Y}}$ and the ground truth $\mathbf{Y}$, formulated as:

$$\text{HuberLoss} = \begin{cases} \frac{1}{2}(\mathbf{Y} - \hat{\mathbf{Y}})^2, & \text{if } \left|\mathbf{Y} - \hat{\mathbf{Y}}\right| \leq \delta \\ \delta \cdot \left(\left|\mathbf{Y} - \hat{\mathbf{Y}}\right| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases} \tag{9}$$

where δ is a positive value.

4 Experiments

4.1 Datasets and Experiment Setup

We evaluate the performance of our proposed Extralonger in long-term and extra-long-term traffic forecasting scenarios using three real-world datasets: PEMS04, PEMS08 and Seattle Loop, which serve as widely adopted benchmarks for traffic prediction tasks. Following early works, we split each dataset into training, validation, and testing sets using a ratio of 6:2:2. [2] Following SSTBAN [4], in

[2] Because of the page limitation, the experiment results of Seattle Loop are in Appendix G.

Table 2: Performance comparison in long-term scenarios. The smaller the value, the better.

Dataset	Mehtod	24 Time Steps			36 Time Steps			48 Time Steps		
		RMSE	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*
PEMS04	HA	81.57	56.47	45.49	106.58	76.01	68.84	127.28	93.37	94.62
	VAR	41.09	27.19	21.42	45.44	30.48	24.51	49.46	33.5	27.28
	DCRNN	42.86	28.70	21.23	51.40	33.78	27.10	57.85	38.26	33.73
	GWNet	35.52	22.79	16.04	38.17	24.71	17.67	40.60	26.42	18.99
	GMAN	38.10	21.67	17.78	52.86	22.12	16.43	47.85	23.35	17.98
	AGCRN	34.44	21.63	14.65	38.19	24.15	16.33	38.26	24.18	16.31
	DMSTGCN	32.09	20.32	14.13	34.86	22.47	15.86	35.05	22.50	16.56
	SSTBAN	32.82	20.17	14.43	34.15	20.82	14.83	35.51	21.66	15.90
	Extralonger	**31.89**	**19.60**	**13.60**	**32.96**	**20.31**	**14.12**	**34.01**	**20.97**	**14.42**
PEMS08	HA	69.72	48.3	32.09	92.72	65.99	46.64	111.85	81.51	61.29
	VAR	44.47	28.31	19.53	48.96	31.7	22.56	52.14	34.51	25.28
	DCRNN	33.34	22.60	15.46	39.37	25.82	18.53	45.64	30.47	25.10
	GWNet	29.47	19.07	12.25	33.54	21.76	13.68	34.20	22.60	14.16
	GMAN	34.29	17.38	15.66	35.89	17.21	16.33	48.54	18.70	16.81
	AGCRN	28.05	17.45	11.25	30.96	19.39	12.73	31.11	19.46	12.88
	DMSTGCN	26.55	16.75	11.44	28.50	18.15	12.64	28.94	18.34	12.93
	SSTBAN	26.32	15.97	12.29	28.30	16.84	12.20	28.82	**16.94**	12.47
	Extralonger	**26.29**	**15.86**	**10.40**	**27.64**	**16.57**	**11.34**	**28.77**	17.14	**11.54**

long-term scenarios, both T and T' are set to 24/36/48. In the extra-long-term scenarios, T and T' are set to 144/288/576/864/1152/1440/1728/2016. Full details are given in Appendix F. We selected three widely used metrics: root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

4.2 Baselines

We carefully choose the following baselines in long-term scenarios. (1) **HA**: Historical Average, which predicts the future traffic flow by averaging the historical data. (2) **VAR** [13]: Vector Auto-Regression, which is a classical multivariate time series forecasting method. (3) **DCRNN** [14]: Diffusion Convolutional Recurrent Neural Network, predicting with diffusion convolutions and GRU. (4) **GWNet** [8]: Graph WaveNet, combining the dilated casual convolutions and graph convolutions to capture dynamics correlation. (5) **GMAN** [15]: Graph Multi-Attention Network, especially using spatial attention with randomly partitioned vertices group. (6) **AGCRN** [16]: Adaptive Graph Convolutional Recurrent Network, which learns node-specific patterns and avoids the pre-defined graphs. (7) **DMSTGCN** [9]: Dynamic and Multi-faceted Spatial-Temporal Graph Convolution Network, which captures the dependency with a spatial learning method and multi-faceted fusion module enhancement. (8) **SSTBAN** [4]: Self-Supervised Spatial-Temporal Bottleneck Attentive Network, utilizing a bottleneck attention scheme to reduce the computational cost.

In extra-long-term scenarios, because the device could not afford the resource cost of the deep learning baselines, we focus on comparing the performance of Extralonger with established statistical methods: HA and VAR.

4.3 Model Implementation and Training Details

Extralonger utilizes the Adam optimizer [17] with the default learning rate of 0.0001, which would decay with predefined milestones. The batch size is set to 16. The hyperparameters remain consistent for different scenarios on the same benchmark, and the layer number L for per route is set to 1. We adopt Huber loss [12] as the loss function, which leverages the strengths of MAE and MSE with $\delta = 1$ while addressing their limitations. The outputs from the three routes (temporal, spatial, and mixed) are combined using a weighted sum, with weights of 0.25, 0.25, and 0.50, respectively. Extralonger is implemented in PyTorch and all experiments were conducted on one single NVIDIA 2080Ti GPU.

4.4 Performance Comparison

The performance results are presented in Table 2 and Table 3 for long-term and extra-long-term scenarios, respectively. The best results are shown in bold.

Table 3: Performance comparison in extra-long-term scenarios. TS: Length of Time Steps.

Dataset	TS	HA			VAR			**Extralonger**		
		RMSE	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*
PEMS04	144	177.11	144.67	220.60	77.22	53.63	64.87	**39.33**	**23.66**	**16.04**
	288	128.50	103.95	166.55	50.43	34.98	34.22	**40.46**	**24.18**	**16.70**
	576	129.35	104.44	166.73	52.90	36.99	35.55	**42.15**	**26.08**	**19.32**
	864	129.65	104.56	167.80	54.55	38.32	37.43	**42.98**	**26.60**	**19.36**
	1152	129.78	104.56	169.59	55.99	39.50	39.12	**42.98**	**26.88**	**19.33**
	1440	129.91	104.66	169.81	56.64	40.05	39.40	**42.30**	**26.62**	**19.24**
	1728	130.00	104.77	169.29	57.01	40.40	39.69	**43.33**	**27.09**	**19.63**
	2016	130.37	105.09	167.48	56.21	39.64	39.23	**43.26**	**27.52**	**20.21**
PEMS08	144	153.98	123.81	114.18	89.00	61.09	46.52	**32.17**	**18.84**	**12.85**
	288	112.80	89.66	87.33	56.07	38.51	26.06	**32.76**	**19.38**	**13.05**
	576	113.80	90.29	87.28	63.16	43.37	28.59	**37.73**	**22.06**	**14.23**
	864	114.36	90.47	87.09	65.07	45.25	30.91	**38.57**	**22.57**	**15.04**
	1152	114.66	90.52	87.67	66.06	45.91	32.42	**40.45**	**23.50**	**16.03**
	1440	114.97	90.80	88.71	67.96	47.03	32.69	**41.44**	**23.78**	**17.90**
	1728	115.39	91.33	89.66	67.36	46.53	32.34	**40.85**	**23.68**	**18.42**
	2016	115.43	91.54	90.01	65.45	44.95	31.37	**41.38**	**23.69**	**19.18**

Our proposed Extralonger outperforms baselines across almost all long-term scenarios. Extralonger consistently surpasses SSTBAN, the second best model, on the PEMS04 dataset, exhibiting an average reduction of 2.36% in RMSE, 2.82% in MAE, and 5.95% in MAPE. Similarly, on PEMS08, Extralonger outperforms SSTBAN by an average margin of 0.87% in RMSE, 0.37% in MAE, and 7.35% in MAPE. The only exception is the MAE in the PEMS08-48 scenario, where Extralonger achieves the second-best result, marginally behind SSTBAN.

In the extra-long-term scenarios, Extralonger demonstrates significant superiority over HA and VAR across all three evaluation metrics. This wide margin in performance shows the effectiveness of Extralonger in capturing extra-long-term temporal dependencies and spatial relationships. In PEMS04, our model delivers steady results when the input time step T is fixed at 288 (a whole day), whereas in PEMS08, the errors rise gradually. This can be attributed to the smaller node count in PEMS08, which enables the model to achieve better performance in smaller-step scenarios.

An interesting observation in extra-long-term scenarios is that, unlike Extralonger obtaining the best performance at $T' = 144$ (length of half a day), the performance of HA and VAR exhibits a decline at $T' = 144$ compared to longer steps. We posit that this decline arises from a potential misalignment between the input and output data. They struggle to predict future trends when the input data contains significantly inverse trends from the predicted future steps. Take a special sample as an example, whose input is 144 steps from the first half of the day and the output is the traffic flow in the second half of the day. It is difficult for HA and VAR to predict the second half solely on the first half.

4.5 Resource Consumption Comparison

Our proposed Extralonger demonstrates superior resource efficiency compared to existing methods, as illustrated in Figure 5. The resource consumption comparison is conducted on the PEMS04 dataset. Due to computational limitations and the high cost of the baseline models, we report actual values (solid lines) for 12, 24, 36, and 48-step predictions. Leveraging the complexity analysis from Section 3.1.1, we extrapolate the resource consumption trends for extra-longer-term scenarios and show the trends using dashed lines in Figure 5.

In long-term scenarios, Extralonger consumes on average 75.87% less memory, 97.13% less training time, and 93.53% less inference time than SSTBAN across 12, 24, 36, and 48-step predictions. The sole exception is memory usage in the 12-step scenario, where the CNN-based method DMSTGCN achieves a marginally lower cost. We speculate that this is because parameter memory usage dominates at the shortest horizon, and DMSTGCN utilizes fewer parameters than our model. However, this advantage diminishes for longer prediction steps (>24 steps), where Extralonger's memory efficiency becomes increasingly pronounced.

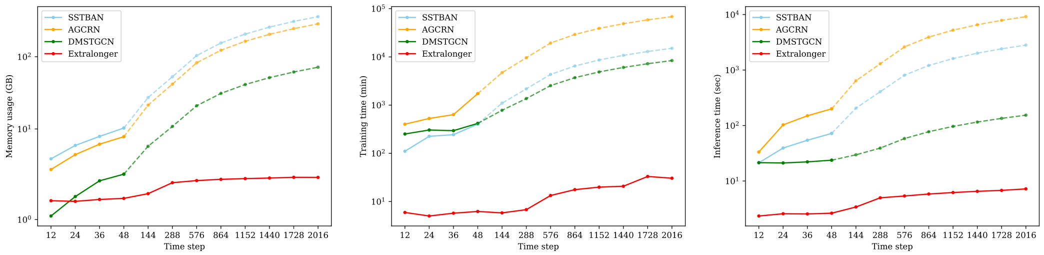


Figure 5: Resource cost comparison w.r.t. memory usage (LEFT), training time (MIDDLE) and inference time (RIGHT). The y-axis represents the logarithm of the value. The dashed line reveals the fitted trendline based on actual results.

Table 4: Ablation study.

Method	RMSE	MAE	MAPE
w/o local & global	37.12	22.69	15.97
w/o global	34.24	21.04	14.45
w/o local	34.82	21.31	14.93
w/o spatial route	37.31	22.72	15.89
w/o temporal route	34.85	21.31	14.30
w/o mixed route	35.59	21.53	14.44
w/o noise injection	34.12	21.03	14.49
Extralonger	**34.01**	**20.97**	**14.42**

In extra-long-term scenarios, Extralonger demonstrates its dominance in resource efficiency. In the 2016-step scenario, our model achieves remarkably low resource consumption, requiring only 2.1 GB of memory, 30.5 minutes for training, and 7.21 seconds for inference. With conservative estimates using fitted polynomial functions, our model's memory usage, training time, and inference time remain only 0.58%, 0.20%, and 0.26% of SSTBAN's cost, respectively. These results strongly support the efficiency gains from the Unified Spatial-Temporal Representation employed by Extralonger.

4.6 Ablation Study

We conducted an ablation study on PEMS04 in the 48-step scenarios. We remove the local part, global part, and global&local part from Global-Local Spatial Transformer, in Table 4. Our findings indicate that both the local and global modules play an important role. Moreover, we explore the necessity of each route in the three-route Transformer. The results show the importance of each route. The ablation of noise injection presents its effectiveness as well.

5 Conclusion

We propose Extralonger, a novel traffic forecasting model that leverages a unified perspective of spatial and temporal factors to address the limitations of current methods in resource efficiency. Extralonger achieves state-of-the-art performance in both long-term and extra-long-term scenarios. Moreover, it demonstrates significant reductions in resource consumption compared to previous models. Specifically, in the longest prediction horizon, memory usage, training time, and inference time are only 0.58%, 0.20%, and 0.26% of the cost of the prior best method, respectively. Furthermore, Extralonger successfully extends the prediction horizon by a remarkable factor of 42 times (from 48 to 2016 steps). We believe that Extralonger establishes a new paradigm for traffic forecasting tasks and paves the way for the development of more efficient models for other spatial-temporal tasks.

6 Acknowledgment

The work described in this paper has been supported by the National Nature Science Foundation of China (No. 62406020).

References

[1] Weiwei Jiang, Jiayun Luo, Miao He, and Weixi Gu. Graph neural network for traffic forecasting: The research progress. *ISPRS International Journal of Geo-Information*, 12(3):100, 2023.

[2] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *Expert systems with applications*, 207:117921, 2022.

[3] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[4] Shengnan Guo, Youfang Lin, Letian Gong, Chenyu Wang, Zeyu Zhou, Zekai Shen, Yiheng Huang, and Huaiyu Wan. Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1585–1596. IEEE, 2023.

[5] Albert Einstein. On the electrodynamics of moving bodies. 1905.

[6] Jianxin Li, Shuai Zhang, Hui Xiong, and Haoyi Zhou. Autost: Towards the universal modeling of spatio-temporal sequences. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 20498–20510. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/80d46bb66ea003f4b29fa6013905d50a-Paper-Conference.pdf`.

[7] Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5, 2023.

[8] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 1907–1913. AAAI Press, 2019. ISBN 9780999241141.

[9] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 547–555, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467275. URL `https://doi.org/10.1145/3447548.3467275`.

[10] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6), dec 2022. ISSN 0360-0300. doi: 10.1145/3530811. URL `https://doi.org/10.1145/3530811`.

[11] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *AAAI*. AAAI Press, 2023.

[12] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964. doi: 10.1214/aoms/1177703732. URL `https://doi.org/10.1214/aoms/1177703732`.

[13] Zheng Lu, Chen Zhou, Jing Wu, Hao Jiang, and Songyue Cui. Integrating granger causality and vector auto-regression for traffic prediction of large-scale wlans. *KSII Trans. Internet Inf. Syst.*, 10:136–151, 2016. URL `https://api.semanticscholar.org/CorpusID:38453985`.

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

[15] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *AAAI*, pages 1234–1241, 2020.

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

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Billy M. Williams and Lester A. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, 129:664–672, 2003. URL `https://api.semanticscholar.org/CorpusID:14712092`.

[19] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1720–1730, 2019.

[20] Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting–full version. *arXiv preprint arXiv:2203.15737*, 2022.

[21] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.

[22] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5415–5428, 2022. doi: 10.1109/TKDE.2021.3056502.

[23] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 3428–3434, 2018.

[24] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[25] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, pages 914–921, 2020.

[26] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 364–373, 2021.

[27] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6367–6374, 2022.

[28] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data*, pages 1–21, 2021.

[29] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. In *Proceedings of the VLDB Endowment*, pages 2733–2746, 2022.

[30] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International Conference on Machine Learning*, pages 11906–11917, 2022.

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

[32] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[33] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems*, 35, 2022.

[34] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=SJU4ayYgl`.

[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJXMpikCZ`.

[36] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=F72ximsx7C1`.

[37] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[38] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4125–4129, 2023.

[39] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

[40] Hermann Minkowski. *Raum und zeit*. Springer, 1988.

[41] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1):96–102, 2001.

[42] Teerath Kumar, Alessandra Mileo, Rob Brennan, and Malika Bendechache. Image data augmentation approaches: A comprehensive survey and future directions, 2023.

[43] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[44] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[45] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction, 2019.

A Related Work

In early works, traffic forecasting is treated as a multivariate time series prediction task, so HA, ARIMA [18], and VAR [13] are the intuitive methods to model it, which neglect the inherent spatial dependencies within traffic data.

With the growing recognition of the equal importance of both spatial and temporal dimensions, traffic forecasting has evolved to consider road networks as graphs where node values (namely, flow, speed or occupancy) change over time. This paradigm shift led to the development of Spatial-Temporal Graph Neural Networks (STGNNs) [3] that leverage GNNs to capture spatial dependencies. STGNNs can be further categorized based on their approach to handling temporal information. RNN-based methods utilize Recurrent Neural Network or its variations to capture temporal dependencies [14, 16, 19]. Attention-based methods [6, 11, 15, 20–23] incorporate attention mechanisms to process temporal information. CNN-based methods [8, 9, 24, 25] can handle spatial and temporal information simultaneously. However, one limitation of CNNs is their restricted receptive window, which can potentially hinder the processing of all nodes across extended temporal horizons. Beyond the aforementioned categories, some studies [26, 27] have focused on spatial information modeling. Additionally, efforts have been made to integrate differential equations with GNNs for enhanced spatial-temporal representation [28–30]. These efforts highlight the ongoing exploration of novel techniques for capturing the intricate interplay between spatial and temporal dynamics in traffic forecasting.

More recently, Transformer-based methods adopted the Transformer architecture to replace GNNs. Vanilla Transformer [31] is introduced for machine translation tasks. Recent years' works have focused on the promising prospect of Transformer-based models [32, 33] to replace the GNNs [34–37] for overcoming the over-smooth issue, which facilitates the evolution of the manner for spatial dimension processing in traffic forecasting field. Variants of Transformer begin to shine in the traffic forecasting field [4, 11, 38, 39].

Most related to ours is the work of AutoST [6], STSGCN [25] and SSTBAN [4]. AutoST and STSGCN notice the line of study that handles time and space separately. The former proposes a universal modeling framework to summarize the aggregation order issue. The latter constructs a new adjacency matrix with pre-defined steps, which limits the model's capability. SSTBAN is the first deep-learning-based work to research long-term traffic forecasting and uses a bottleneck attention scheme to reduce the computational cost to $\mathcal{O}(TN)$, but it has quadratic complexity on memory usage as well.

Our work points out the deficiency in the classical representation of spatial-temporal systems, which results in high resource costs. We propose the Unified Spatial-Temporal Representation to unify the spatial and temporal dimensions to reduce the time and space complexity without any architectural modification, and successfully extend the prediction horizon to one week, following in the footsteps of SSTBAN.

B Inspiration from Relativity

The Unified Spatial-Temporal Representation is inspired by the theory of relativity, a theory in physics that explains the universal, general relationship between space and time. Similarly, traffic forecasting inherently deals with uncovering the intertwined nature of spatial and temporal dimensions in traffic data. This realization motivated us to explore seemingly disparate fields to find solutions, mirroring the pursuit of knowledge across disciplines that foster scientific progress.

Space and time are unified and inseparable in the relative view. In other words, when referring to one particular time step, we should also consider the spatial information, and vice versa. This translates directly to the context of traffic forecasting: understanding a specific time step necessitates simultaneous consideration of the corresponding spatial information, and vice versa. As the famous verb of Hermann Minkowski [40], "space for itself, and time for itself shall completely reduce to a mere shadow, and only some sort of **union** of the two shall preserve independence". This implies that we should address both time and space concurrently (Unified Spatial-Temporal Representation), rather than isolating them (classical representation).

Thus, we propose the Unified Spatial-Temporal Representation to unify the spatial and temporal dimensions. As an additional explanation of the main body analysis, the D in temporal representation $\mathbf{E}_t$ has spatial information, and the D in spatial representation $\mathbf{E}_s$ has temporal information, which is the essence of the Unified Spatial-Temporal Representation. In this sense, we use the spatial information to define the time step and the temporal information to define the spatial node. From a data representation perspective, the data representation is 2D with Unified Spatial-Temporal Representation and 3D with classical representation.

C Complexity Analysis

The high-complexity issue in classical representation (as depicted in Equation 2 and Figure 2) hinders the prediction horizon extension. In the main body of this paper, we have concisely discussed the time and space complexity. We will give the full analysis in this section. We omit the batch size B. Focusing on the difference between the classical representation and the Unified Spatial-Temporal Representation, we focus on the core architecture with one single layer rather than delving into the details of each compared model. This setting allows for a more controlled comparison, highlighting the specific contribution of the proposed representation to the overall efficiency of the model.

Time Complexity We first analyze methods using the classical representation. For RNN-based methods, due to the repetitive operation for the spatial and temporal dimensions, the time complexity increases to $\mathcal{O}(TN)$.

For attention-based methods, for example, Transformer architecture, the time complexity grows by one order to $\mathcal{O}(NT^2 + TN^2)$. In attention-based and Transformer-based methods, to simplify the analysis of various derivative models, we primarily concentrate on the basic self-attention mechanism for temporal and spatial dimensions, whose time complexity grows by one order to $\mathcal{O}(NT^2 + TN^2)$. The time complexity of CNN-based methods is influenced by the feature map size, kernel size and the number of channels. In our setting, the channel size corresponds to D. For fairness, we omit this part and only consider the feature map size and kernel size. We use the biggest feature map size, i.e., $T \times N$. Therefore, the time complexity is $\mathcal{O}(k^2TN)$, where k is the kernel size. While the complexity of GNN is linear, the total complexity is $\mathcal{O}(k^2TN + TN)$.

When using the Unified Spatial-Temporal Representation, because we adopt the Transformer architecture in Extralonger, the time complexity is $\mathcal{O}(T^2 + N^2)$ for the self-attention mechanism.

Space Complexity As mentioned in the main body, we need to consider many factors in space complexity, such as the parallel framework, optimizer type, and whether to maintain the intermediate states, which is out of our scope. Therefore, we simplify it. Besides, we omit the parameter memory usage M_{model}, because we focus on the impact of data instead of the parameter memory usage for extra-long-term scenarios.

The space complexity of RNN-based methods and self-attention mechanisms scales linearly with the sequence length T and the number of nodes N due to the repetitive computations involved. These repetitive operations effectively act like a batch size and exhibit an approximately linear correlation with memory consumption. Consequently, the space complexity for RNN-based with classic representation is $\mathcal{O}(TN)$, while for attention-based and Transformer-based methods is $\mathcal{O}(NT^2 + TN^2)$.

For CNN-based methods, the original space complexity (ignoring the parameter memory usage) is typically $\mathcal{O}(Height \times Width)$, which can be high depending on the input data size. In the context of traffic forecasting, the space complexity is $\mathcal{O}(TN)$ as well.

With the Unified Spatial-Temporal Representation, the space complexity is $\mathcal{O}(T^2 + N^2)$ in Extralonger, which is a one-order reduction compared with the classical representation.

D Why is Global-Local Spatial Transformer

Global Part The GNN-based methods capture spatial dependency using static prior topology information, hypothesizing that the traffic flow trends among adjacent nodes are similar. However, we observe that non-adjacent nodes, such as Node 95 and 111, exhibit similar flow patterns. This suggests that dependencies can exist between nodes that are not spatial neighbors. To capture these long-range dependencies, Extralonger leverages a Transformer architecture with full connectivity between all nodes. With the Transformer, we treat the traffic road network as a fully connected graph, thus the α_{global} is gained. This global design enables Extralonger to go beyond the limitations of local neighborhoods and directly model relationships between any two nodes in the road network, regardless of their spatial proximity. The belief in learning global correlations is also supported by studies such as Guo et al. [4], Wu et al. [8], Zheng et al. [15].

Local Part While the proposed model captures long-range dependencies through the fully connected Transformer, the road network topology, represented by the adjacency matrix, remains a valuable source of prior knowledge. This matrix encodes the inherent spatial relationships between nodes, providing information about the correlations in traffic flow between adjacent nodes. In essence, the model leverages the adjacency matrix to complement the spatial dependency modeling capabilities of the Transformer. Specifically, the adjacency matrix is injected as a mask to obtain the α_{local} due to they inherently have the same dimensionality. The local design draws inspiration from the works of Dwivedi and Bresson [32] and Rampášek et al. [33]. Refer to the Section 3.2.2 for other procedures.

E Learnable Noise Injection

Caltrans Performance Measurement System [41] and SSTBAN [4] both claim the importance of mitigating the impact of noise on the traffic forecasting model's performance. They acknowledge the inherent challenge of data validity in traffic forecasting. Data augmentation technique is a possible way to improve robustness. However, traditional data augmentation techniques commonly used in computer vision (CV) [42], such as cropping, scaling, and rotation, are not well-suited for traffic data due to their inherent sequential nature and strong self-correlations. Additionally, complex deep learning methods like VAEs [43] and GANs [44] might be overly elaborate for this specific task.

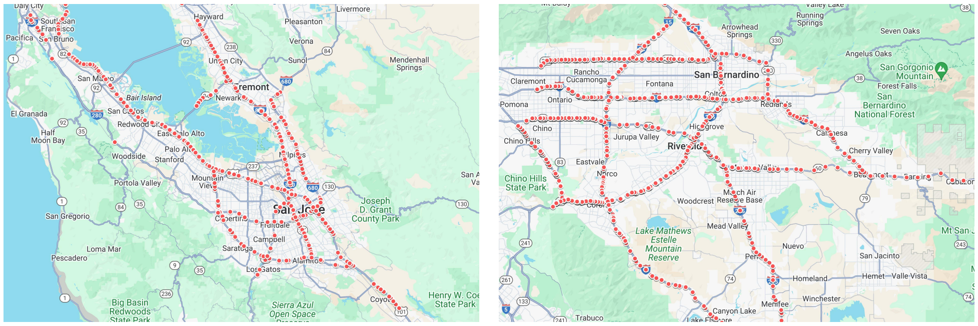
To mitigate the impact of noise, we propose a simple yet effective data augmentation strategy – involving the injection of learnable noise into the input data, which is implemented by the learnable parameter initialized with Xavier uniform distribution. We use this learnable noise embedding to imitate the impulse noise and enhance the model's robustness against real-world noise patterns. The result of the ablation experiment validates the effectiveness of this technique (Section 4.6).

F Details of Datasets and Setup

Both PEMS04 and PEMS08 datasets are collected from the Caltrans Performance Measurement System [41] at a 5-minute sampling frequency. Seattle Loop dataset is collected by the inductive loop detectors deployed on freeways in the Seattle area [45]. PEMS04 contains 307 nodes and 16992 time steps, covering the time range from 2018/01 to 2018/02. PEMS08 contains 170 nodes and 17856 time steps, covering the time range from 2016/07 to 2016/08. The Seattle Loop dataset consists of 323 nodes and 8,760 time steps, ranging from 2015/01 to 2015/12. To ensure a fair comparison, the setup of Seattle Loop follows SSTBAN [4] that aggregates the original 5-minute granularity to a 1-hour interval. The detailed information is summarized in Table 5. Moreover, we illustrate the node position of PEMS04 and PEMS08 with the latitude and longitude in Figure 6. Following SSTBAN [4], we normalize the data with the Z-score method and the mean and standard deviation are calculated from the training set.

In the long-term scenarios, the $T - T'$ is symmetric, and we set them as 24, 36, and 48 steps. In the extra-long-term scenarios, we set one scenario as 144-144 for half of the day prediction. For cases where the future prediction step T' exceeds one day, T is fixed to 288, and the future prediction step increases from 1 day to 7 days with a daily increment. This configuration considers a whole day as input, as it encompasses a complete daily cycle.

We slide the window to gain the input data $\mathbf{X}$ and ground truth $\mathbf{Y}$. We stride one step for each sample, and the length corresponds to the T and T'.



(a) PEMS04

(b) PEMS08

Figure 6: Node position of PEMS04 and PEMS08.

Table 5: Summary of Datasets.

Dataset	# Time Steps	# Nodes	Time Range
PEMS04	16992	307	2018/01 - 2018/02
PEMS08	17856	170	2016/07 - 2016/08
Seattle Loop	8760	323	2015/01 - 2015/12

Table 6: Performance comparison on long-term traffic speed forecasting in Seattle Loop.

Time step	24			36			48		
Method	*RMSE*	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*
HA	11.86	8.07	26.57	12.37	8.47	27.76	12.31	8.49	27.82
VAR	9.56	6.21	19.94	9.96	6.44	21.28	10.28	6.69	22.24
DCRNN	7.97	4.37	14.04	8.38	4.60	14.41	8.63	4.73	14.91
GWNet	7.84	4.28	14.06	8.18	4.60	15.12	8.35	4.67	15.04
GMAN	7.84	4.13	12.88	8.10	4.23	12.95	8.09	4.26	13.26
AGCRN	7.83	4.27	13.53	8.31	4.66	14.76	8.60	4.82	15.62
DMSTGCN	7.59	4.08	13.51	7.98	4.31	14.31	8.20	4.49	14.86
SSTBAN	7.72	4.05	12.69	7.83	4.11	12.44	7.88	4.12	12.25
Extralonger	**7.43**	**4.04**	**12.48**	**7.51**	**4.05**	**11.96**	**7.68**	**4.11**	**12.04**

Table 7: Performance comparison on extra-long-term traffic speed forecasting in Seattle Loop.

Method	HA			VAR			**Extralonger**		
Time step	*RMSE*	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*
144	11.95	8.30	27.48	11.17	7.33	24.75	**8.07**	**4.33**	**12.64**
288	12.05	8.33	28.04	11.09	7.31	24.14	**8.20**	**4.44**	**14.03**
576	11.95	8.14	27.88	11.30	7.43	24.83	**8.15**	**4.59**	**14.46**
864	11.90	8.16	27.57	11.42	7.49	25.33	**8.07**	**4.50**	**13.63**
1152	11.87	8.02	27.43	11.53	7.54	25.77	**8.00**	**4.39**	**12.94**
1440	11.85	7.94	27.23	11.59	7.58	26.01	**7.69**	**4.15**	**12.36**
1728	11.83	7.88	26.96	11.70	7.65	26.49	**7.66**	**4.11**	**12.00**
2016	11.79	7.85	26.71	11.76	7.71	26.78	**7.57**	**4.02**	**11.63**

G Experiments of Seattle Loop

The results in the Seattle Loop dataset are presented in Table 6 and Table 7 for long-term and extra-long-term scenarios, respectively. The best results are shown in bold. Our model Extralonger outperforms the baseline models for all metrics in all scenarios. Interestingly, in the extra-long-term scenarios, the prediction from Extralongerbecomes even better when the step length increases. We speculate that this improvement can be attributed to the increase in data volume, which allows the model to capture and learn long-range dependencies that may not be apparent in the shorter step length scenarios. The larger amount of data provides a richer context for the model to extract patterns and make more accurate predictions.

H Experiments of Error Bar

We conducted error bar comparison experiments using box plots on PEMS04 and PEMS08 to compare our model with the prior best model, SSTBAN [4]. We analyzed six sets of results. As shown in Figure 7 and Figure 8, our Extralonger consistently outperforms SSTBAN, as evidenced by smaller medians and shorter interquartile ranges. These findings indicate that our model demonstrates steady improvement over the prior best model and exhibits lower variability.

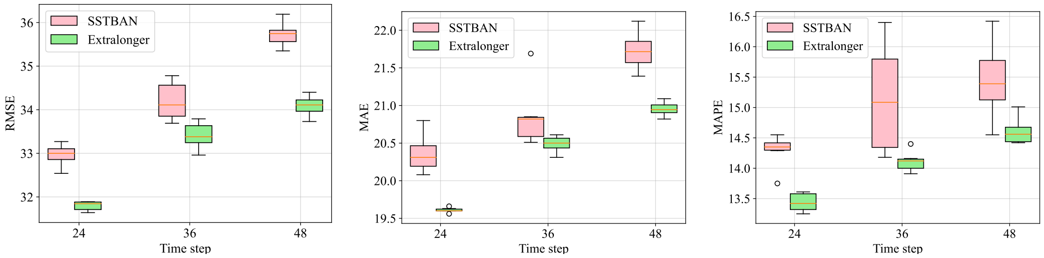


Figure 7: Error bar on PEMS04 w.r.t RMSE (LEFT), MAE (MIDDLE) and MAPE (RIGHT).

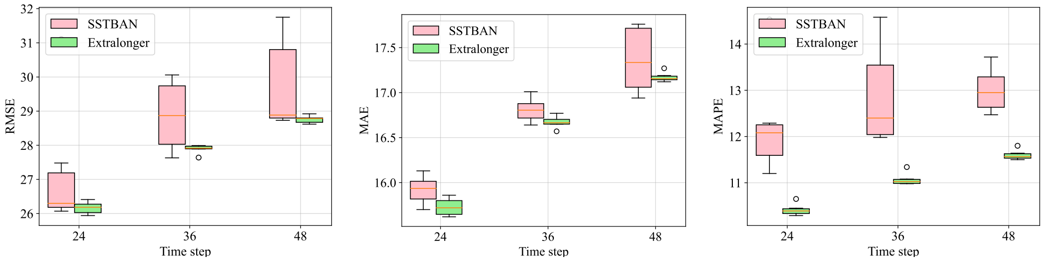


Figure 8: Error bar on PEMS08 w.r.t RMSE (LEFT), MAE (MIDDLE) and MAPE (RIGHT).

I Broader Impact

Our work could have a broader impact beyond traffic forecasting. It has the potential to significantly advance the field of transportation and potentially influence other general spatial-temporal tasks. Here are some key areas where our work could have a significant impact.

Transportation Field

- **Improved Traffic Management:** By enabling accurate extra-long-term traffic forecasting, Extralonger can empower traffic management authorities to proactively optimize traffic flow, reduce congestion, and improve overall transportation efficiency.
- **Resource-Constrained Applications:** The remarkable reduction in resource consumption achieved by Extralonger unlocks the possibility of deploying traffic forecasting models on devices with limited computational power. This opens doors for real-time traffic prediction in resource-constrained environments, such as embedded systems in vehicles or edge computing devices.

General Spatial-Temporal Tasks
The success of Extralonger's unified spatial-temporal representation suggests a potential paradigm shift in approaching other tasks involving spatial and temporal dependencies. This Unified Spatial-Temporal Representation could be explored and adapted to various domains requiring accurate long-horizon prediction based on interconnected spatial and temporal data, for example:

- **Weather Forecasting:** Weather forecasting could benefit from a unified representation of spatial (e.g., atmospheric pressure, temperature) and temporal (e.g., historical weather patterns) data for improved long-term prediction.
- **Urban Planning:** Urban planning could utilize models enhanced by Unified Spatial-Temporal Representation to forecast future resource demands (energy, water) based on spatial distribution and historical trends, enabling more sustainable infrastructure development.

Overall, Extralonger represents a significant advancement in traffic forecasting and paves the way for a more efficient approach to tackling various spatial-temporal modeling challenges.

J Limitations

While Extralonger has successfully tackled the high-complexity issues inherent in classical representation, achieving a one-week prediction horizon, it still exhibits two certain limitations.

Extralonger effectively captures the dominant trend patterns in traffic flow data. However, Extralonger's performance exhibits some degradation when encountering significant fluctuations, particularly during peak traffic hours (9:00-15:00). Real-world traffic data inherently exhibits these rapid variations, also known as pulse variations, which remain a challenge for current traffic forecasting models. This limitation presents an exciting avenue for future research, aiming to develop more robust methods for capturing and predicting such short-term fluctuations.

Being fundamentally data-driven, our model relies solely on historical data for predictions and thus lacks the capability to respond to emergent traffic activities. This limitation may lead to potential misinterpretations within the ITS. Therefore, it is imperative to integrate additional information when planning and managing the traffic system.

K Future Work

It turns out that prediction errors tend to increase during periods of significant traffic flow fluctuations, particularly during peak hours (9:00-15:00). Real-world traffic data inherently exhibits pulse variations, which remain a challenging forecasting issue. We acknowledge this limitation and identify it as a promising avenue for future research.

Moreover, building upon the success of the Unified Spatial-Temporal Representation in Extralonger, we aim to explore its applicability to a broader range of spatial-temporal tasks. This would demonstrate the generalization and efficiency of our proposed method, potentially establishing it as a versatile tool for various spatial-temporal tasks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See the abstract and introduction part.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The forecasting error rises during 9:00-15:00, the peak traffic hours.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The complexity reduction theoretical analysis is based on some assumptions. We give the details in Section 3.1.1 and Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We introduce our method, Extralonger, step by step (Section 3 and the detailed implementation is given in Section 4.3. The setup of experiments and the datasets are detailed in Appendix F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the code and data are available in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test datasets partitioning details are given in Appendix F. The hyperparameters and optimizer are detailed in Section 4.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the RMSE, MAE and RMSE in long-term and extra-long-term scenarios (Section 4.4), and we also gain a great reduction in training time, inference time and memory usage (Section 4.5).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We conducted our experiment on one single NVIDIA 2080Ti GPU. The implementation details are given in Section 4.3 and the experiment package dependency is in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have checked the code of ethics, and our research is in line with the guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix I

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our research is for traffic forecasting, which does not have the misuse risk mentioned in the question.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the code we use is from open-source libraries. The datasets are the public dataset. And we cited them properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should citep the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code is available in the supplementary material and we give a simple README.md file to introduce the launch commands.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our research does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.