A COMPLEXITY ANALYSIS OF SPATIAL AND TEMPORAL MODULES

In our analysis, it is assumed that the intermediate features are represented by $\mathbf{H} \in \mathbb{R}^{N \times T \cdot d}$. The complexities of the temporal modules and spatial modules are comprehensively analyzed in Sections A.1 and A.2, respectively.

Table 5: Summary of Key Notations

Notation	Description
$\mathbf{X}$	Input matrix of incomplete CTS input
$\mathtt{N}$	Total number of time series in $\mathbf{X}$
$\mathtt{T}$	Window size of the CTS
$\mathtt{d}$	Embedding size of the embedded CTS
$\mathtt{L}$	Number of feature extraction modules

A.1 Temporal Module

CNN-based temporal modules, with the Temporal Convolutional Network (TCN) as a key example, represent a significant advancement in processing time series data. TCNs distinguish themselves from standard CNNs through the use of causally ordered convolutions, ensuring that outputs are derived solely from past and present inputs, without affecting computational complexities.

For a given input time series data $\mathbf{X} \in \mathbb{R}^{\mathtt{N} \times \mathtt{T}}$, transformed into a latent representation $\mathbf{H} \in \mathbb{R}^{\mathtt{N} \times \mathtt{T} \times \mathtt{d}}$, a TCN layer operates as follows:

$$\begin{aligned} &\mathrm{TCN}(H \mid \delta, \mathtt{K}) = H', \text{ where} \\ &H'[i;t;d] = \sum_{k=0}^{\mathtt{K}-1} \left(H[i; t - \delta \cdot k; :] \cdot \mathrm{W}^d[k;:] \right), \end{aligned}$$

where $\mathrm{W}^d \in \mathbb{R}^{\mathtt{d} \times \mathtt{K}}$ represents the d-th convolutional kernel, with $\mathtt{K}$ and δ denoting the kernel size and dilation rate, respectively.

CNN-based Temporal Modules

Space Complexity: The TCN employs $\mathtt{d}$ kernels, each of size $\mathtt{K} \times \mathtt{d}$, leading to a space complexity of $O(\mathtt{K} \cdot \mathtt{d}^2)$. Given that $\mathtt{K}$ is relatively small and considered a constant, the space complexity simplifies to $O(\mathtt{d}^2)$. With $\mathtt{L}$ modules, the overall space complexity is $O(\mathtt{L} \cdot \mathtt{d}^2)$.
Time Complexity: The time complexity for a single convolution operation involving one kernel is $O(\mathtt{d})$. Extending this across $\mathtt{T}$ temporal convolutions for $\mathtt{N}$ nodes with $\mathtt{d}$ kernels results in a time complexity of $O(\mathtt{d}^2 \cdot \mathtt{N} \cdot \mathtt{T})$. Therefore, with $\mathtt{L}$ such modules, the overall time complexity is $O(\mathtt{L} \cdot \mathtt{d}^2 \cdot \mathtt{N} \cdot \mathtt{T})$.

RNN-based temporal modules, particularly LSTMs (Long Short-Term Memory) and GRUs (Gate Recurrent Units), are pivotal in capturing temporal dependencies in time series data. These modules are distinguished by their gate mechanisms, which efficiently regulate the flow of information through the network. Despite the difference in the number of gates between LSTMs and GRUs, this constant does not affect their complexities.

The operations of an LSTM at each timestamp t are given by:

$$\begin{aligned} i_t &= \sigma\left(\mathrm{W}_{ii} H(i)_t + \mathrm{W}_{hi} h_{t-1}\right), \\ f_t &= \sigma\left(\mathrm{W}_{if} H(i)_t + \mathrm{W}_{hf} h_{t-1}\right), \\ g_t &= \tanh\left(\mathrm{W}_{ig} H(i)_t + \mathrm{W}_{hg} h_{t-1}\right), \\ o_t &= \sigma\left(\mathrm{W}_{io} H(i)_t + \mathrm{W}_{ho} h_{t-1}\right), \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\ h_t &= o_t \odot \tanh\left(c_t\right), \end{aligned} \tag{21}$$

where $H(i)_t$ is the i-th time series features at the t-th timestamp of $\mathbf{X}$, h_t is the hidden state at the t-th timestamp of LSTM, and i_t, f_t, g_t, o_t denote the outputs of the input gate, the forget gate, the cell update, and the output gate, respectively.

RNN-based Temporal Modules

Space Complexity: The space complexity of an LSTM-based temporal module is derived from the weight matrices associated with its gates, yielding $O(\mathtt{H_h} \cdot \mathtt{d} + \mathtt{H_h}^2)$. Simplifying by equating the size of the hidden state $\mathtt{H_h}$ with the embedding size $\mathtt{d}$, the space complexity becomes $O(\mathtt{d}^2)$, independent of the number of series $\mathtt{N}$ and timestamps $\mathtt{T}$.
Time Complexity: The time complexity for processing an input at a single gate, as illustrated in Equation 21, is $O(\mathtt{H_h} \cdot (\mathtt{H_h} + \mathtt{d}))$, which simplifies to $O(\mathtt{d}^2)$. Considering $\mathtt{L}$ LSTM modules processing $\mathtt{N}$ time series each of length $\mathtt{T}$, the overall time complexity is $O(\mathtt{L} \cdot \mathtt{d}^2 \cdot \mathtt{N} \cdot \mathtt{T})$.

Self-attention-based temporal modules, notably characterized by Multi-Head Attention (MHA) and Feed-Forward Network (FFN) layers, excel in capturing complex dependencies within time series data. The MHA layer leverages parallel self-attention heads to transform input embeddings H^{PE} through learnable matrices $\mathrm{W}_i^Q, \mathrm{W}_i^K, \mathrm{W}_i^V$, computing attention scores that enable a nuanced aggregation of temporal features. Coupled with the FFN layer, which introduces non-linearity and dimensionality transformations, these components form the core of the self-attention mechanism.

Self-attention-based Temporal Modules

Space Complexity: The space complexity of self-attention-based temporal modules is $O(\mathtt{d}^2)$, accounting for 3h projection matrices in the MHA layer and two linear transformations in the FFN layer, with the embedding size $\mathtt{d_F}$ assumed proportional to $\mathtt{d}$. This complexity remains independent of the number of time series $\mathtt{N}$ and timestamps $\mathtt{T}$, as parameters are shared across inputs.
Time Complexity: The overall time complexity is $O(\mathtt{L} \cdot \mathtt{N} \cdot \mathtt{d} \cdot \mathtt{T} \cdot (\mathtt{T} + \mathtt{d}))$. This arises from:

- The MHA layer's operations, which involve linear projections and the self-attention mechanism scaling with $\mathtt{T}^2$ due to pairwise interactions among timestamps, contributing $O(\mathtt{d}^2 \cdot \mathtt{N} \cdot \mathtt{T})$ to the complexity.

- The FFN layer's two linear transformations, which also contribute to the time complexity, reinforcing the model's capacity to process and transform temporal data efficiently.

A.2 Spatial Module

Graph Convolutional Networks (GCN) S-modules are essential in capturing spatial dependencies within graph-structured data. These modules perform two pivotal steps: neighborhood information aggregation and subsequent nonlinear transformation. The aggregation step involves the use of a normalized adjacency matrix $\mathsf{A} \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$ to blend features from neighboring nodes, followed by a convolution operation that applies a weight matrix $W \in \mathbb{R}^{\mathsf{d}\times\mathsf{d}}$ to the aggregated features, formulated as $GCN(H_G) = \sigma(A \cdot H_G \cdot W)$.

GCN-based Spatial Modules

Space Complexity: The space complexity is directly associated with the GCN weight matrices across L modules, resulting in $O(\mathsf{L}\cdot\mathsf{d}^2)$. This calculation does not include the number of nodes N or timestamps T, under the assumption of shared parameters across the graph's features.
Time Complexity: For a single timestamp, the time complexity encompasses matrix multiplication between the adjacency matrix and node features, followed by the application of the weight matrix. This complexity is $O(\mathsf{N}^2\cdot\mathsf{d}+\mathsf{N}\cdot\mathsf{d}^2)$, accounting for both the aggregation and transformation steps. Extending these operations across L modules and T timestamps, the overall time complexity for the GCN-based S-module is $O(\mathsf{L}\cdot\mathsf{N}\cdot\mathsf{d}\cdot\mathsf{T}\cdot(\mathsf{N}+\mathsf{d}))$.

Self-attention-based S-modules, mirroring their temporal counterparts, employ Multi-Head Attention (MHA) and Feed-Forward Network (FFN) layers to capture spatial correlations across nodes over time. Unlike temporal modules that partition input features by time slices, spatial modules divide them into N slices, each representing all timestamps for a single node, hence focusing on the interactions among nodes within the entire time series.

Self-attention-based Spatial Modules

Space Complexity: The space complexity for these modules remains consistent with the temporal analysis, dictated by the $3\mathsf{h}$ linear projection matrices in the MHA and two FFN layers' weight matrices, culminating in $O(\mathsf{L}\cdot\mathsf{d}^2)$. This calculation assumes shared parameters across nodes and timestamps, thus independent of both N and T.
Time Complexity: However, adjusts to account for the spatial correlations among N nodes over T timestamps. The linear projections contribute $O(\mathsf{d}^2\cdot\mathsf{N}\cdot\mathsf{T})$, while the self-attention mechanism, focusing on node interactions, incurs an additional $O(\mathsf{N}^2\cdot\mathsf{d}\cdot\mathsf{T})$. The FFN layer, implementing two linear transformations, similarly affects the time complexity, maintaining the overall computation at $O(\mathsf{L}\cdot\mathsf{d}^2\cdot\mathsf{N}\cdot\mathsf{T}+\mathsf{N}^2\cdot\mathsf{d}\cdot\mathsf{T})$. Simplified, the total time complexity is expressed as $O(\mathsf{L}\cdot\mathsf{N}\cdot\mathsf{d}\cdot\mathsf{T}\cdot(\mathsf{N}+\mathsf{d}))$, illustrating the balance between spatial detail resolution and computational demands.

B IMPLEMENTATION DETAILS

Experiments were conducted using a high-performance computing setup featuring an NVIDIA Quadro RTX 8000 GPU and an Intel Xeon Gold 5215 CPU at 2.50GHz.

In the implementation of the MvLC mechanism, embedding sizes were configured as follows: for the AQ36 and PeMS series datasets, the sizes for $\mathbf{CB}^{\mathrm{PT}}$, $\mathbf{CB}^{\mathrm{PS}}$, and $\mathbf{CB}^{\mathrm{PST}}$ were set to 8, 8, and 16, respectively, with raw CTS embeddings at 16, and $\mathbf{E}^{\mathrm{PPE}}$'s embedding size at 32. For the COVID-19 dataset, embedding sizes for $\mathbf{CB}^{\mathrm{PT}}$ and $\mathbf{CB}^{\mathrm{PS}}$ were adjusted to 24, $\mathbf{CB}^{\mathrm{PST}}$ to 48, raw CTS embedding to 48, and $\mathbf{E}^{\mathrm{PPE}}$ to 64. The AQ36 and COVID-19 datasets each employed one CaA module, while the PeMS series utilized two, with the group numbers for GFFNs configured to 1 for AQ36 and COVID-19, and 2 for the PeMS series. An Adam optimizer with a learning rate of 0.0002 was used across 400 epochs for thorough training and optimization of the models.

Further implementation specifics, including additional configuration parameters and datasets, are documented within the anonymous codebase [1], accessible for review and replication purposes.

C ABLATION STUDY

The experiments in this section are the supplement to those in Section 5.3 (on Page 8) in the submission. We report the results of the ablation study on the COVID-19 dataset, and the PeMS series datasets in Tables 6, and 7, respectively.

Table 6: Ablation study of ReCTSi on COVID-19.

Model	FLOPs (M)	Params (K)	Peak Mem (MB)	Latency (ms)	MAE	MSE	MRE
ReCTSi\PT	0.10	126	8.88	1.70	0.023	0.002	0.002
ReCTSi\PS	0.10	125	8.98	1.68	0.019	0.001	0.001
ReCTSi\PST	0.08	83	8.16	1.63	0.029	0.003	0.004
ReCTSi\CF	0.37	411	25.49	1.75	0.021	0.002	0.002
ReCTSi\CM	0.12	145	9.81	1.76	0.020	0.002	0.002
ReCTSi\SA	0.07	92	5.19	1.49	0.028	0.003	0.004
ReCTSi\TA	0.07	94	5.19	1.52	0.036	0.003	0.004
ReCTSi	0.12	145	9.81	1.76	**0.015**	**0.000**	**0.001**

Table 7: Ablation study of ReCTSi on PeMS series datasets.

Model	Efficiency Metrics				PeMS-BA			PeMS-LA			PeMS-SD		
	FLOPs (M)	Params (K)	Peak Mem (MB)	Latency (ms)	MAE	MSE	MRE	MAE	MSE	MRE	MAE	MSE	MRE
ReCTSi\PT	0.05	74	11.67	1.74	19.116	1066.450	0.047	21.760	1575.920	0.041	16.123	800.321	0.041
ReCTSi\PS	0.05	74	11.77	1.72	20.337	1107.545	0.050	22.100	1603.884	0.041	15.859	767.537	0.040
ReCTSi\PST	0.04	44	11.01	1.68	20.931	1170.524	0.051	22.839	1657.482	0.042	16.955	859.840	0.042
ReCTSi\CF	0.09	113	15.60	1.80	19.597	1065.052	0.047	21.552	1568.404	0.040	16.402	810.457	0.041
ReCTSi\CM	0.06	83	12.54	1.78	19.832	1075.374	0.048	21.467	1525.109	0.040	15.196	731.374	0.039
ReCTSi\SA	0.04	63	6.57	1.53	24.867	1584.684	0.060	23.753	1699.924	0.044	18.291	899.502	0.046
ReCTSi\TA	0.04	64	6.57	1.49	26.485	1733.853	0.069	25.101	1800.761	0.051	19.557	950.672	0.047
ReCTSi	0.06	83	12.54	1.79	**18.998**	**1050.328**	**0.046**	**21.130**	**1469.592**	**0.040**	**15.024**	**721.439**	**0.037**