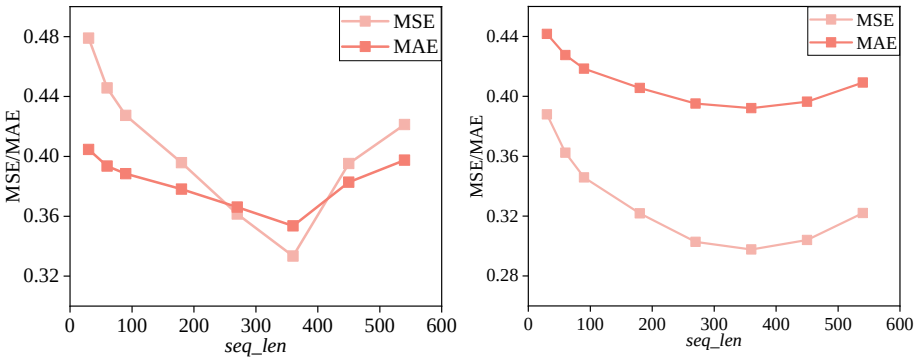
Supplementary Materials

June 16, 2025

A The impacts of parameter on the performance of STDMamba

A.1 The impacts of parameter seq_len on the performance of STDMamba

Concretely, we evaluate 8 different input lengths *seq_len*, i.e., {30, 60, 90, 180, 270, 360, 450, 540}, to determine the most suitable *seq_len* for STDMamba. According to the results in Figure 1, the best prediction performance is achieved when the input length is set to 360, which aligns with the seasonal characteristics of the SST variations and the cyclic nature of SST temporal dynamics since one year covers the complete cycle of SST changes. If the lookback window exceeds one year, the periodic nature of SST may lead to redundant information. Therefore, we set the *seq_len* to 360 for all prediction horizons on both datasets.

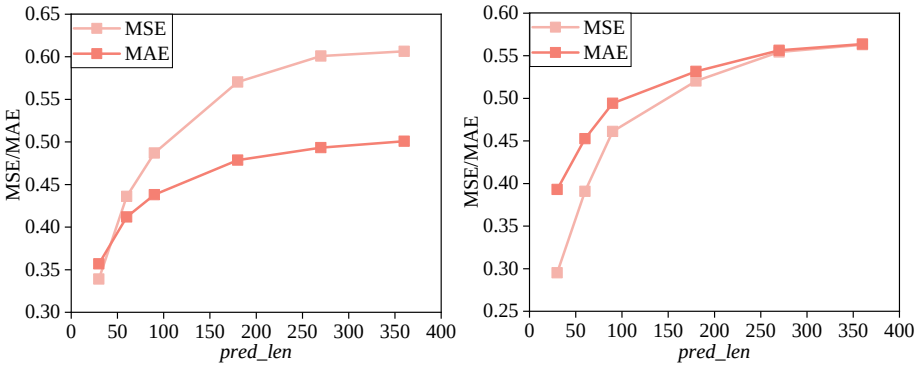


((a)) NPO-SST ((b)) INO-SST

Figure 1: The results of STDMamba on the NPO-SST and INO-SST datasets while varying input lengths, where the prediction length is 30.

A.2 The impacts of parameter pred_len on the performance of STDMamba

In addition, we fix the input length to 360 to investigate the effect of output length on predictive performance. As shown in Figure 2, the prediction accuracy of STDMamba gradually declines with the increase of prediction length. This trend is not unique to our STDMamba, and similar performance degradation is observed across all state-of-the-art baseline models, indicating that long-term forecasting is a general challenge in time series prediction.



((a)) NPO-SST ((b)) INO-SST

Figure 2: The results of STDMamba on the NPO-SST and INO-SST datasets while varying prediction length, where the input length is 360.

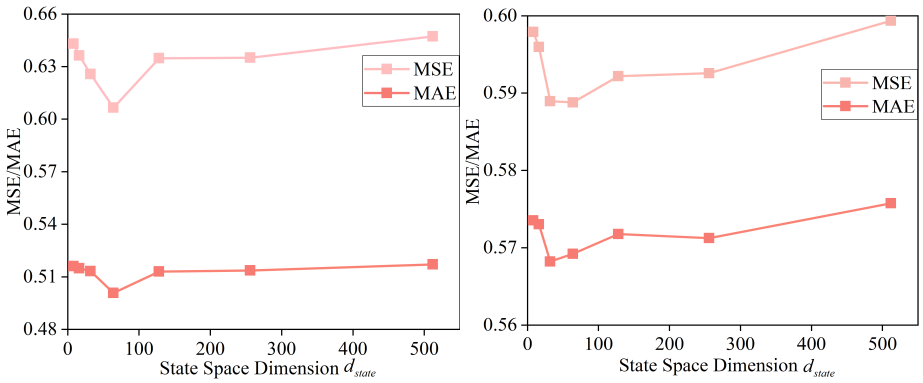
A.3 The impacts of parameter d_state of Mamba2 on the performance of STDMamba

We determine the optimal values for the hyper-parameter, including the state space dimension d_{state} of Mamba2 and the kernel size K of the Gaussian convolution, for STDMamba through experiments.

The parameter d_{state} in Mamba2 determines the model's memory capacity and representation complexity. An excessively high dimension may introduce noise and lead to overfitting, whereas an excessively low dimension may hinder the model's ability to capture essential features, resulting in underfitting. Figure 3 shows the results of STDMamba for different d_{state}, i.e., {8, 16, 32, 64, 128, 256, 512}. Based on these results, we set d_{state} to 64 for both datasets.

A.4 The impacts of parameter kernel size K of the Gaussian convolution the performance of STDMamba

To reduce the noise in long-term fine-grained SST prediction, we select a large kernel size K to better capture long-term dependencies. We evaluated 11 dif-



((a)) NPO-SST ((b)) INO-SST

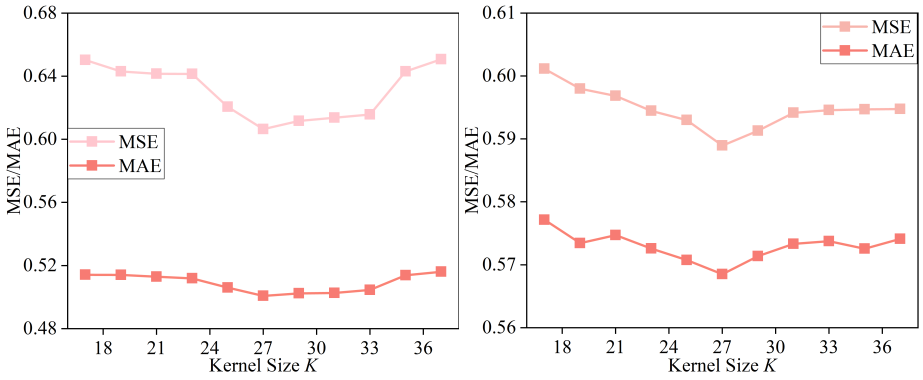
Figure 3: The results of STDMamba on the NPO-SST and INO-SST datasets while varying parameter d_{state}, where the prediction length is 360.

ferent large kernel sizes K, i.e., {17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37}, to identify the most suitable K for the STDMamba model. Based on the results presented in Figure 4, we set the kernel size K to 27 for two datasets.

B Prediction Efficiency Analysis

we provide a detailed discussion on the computational complexity of the proposed STDMamba model and compare it with multiple baseline methods in terms of training time, inference time, number of parameters, and prediction performance.

For evaluating the computational complexity of STDMamba, We assume the input has a shape of (B, N, d_{model}), where B denotes the batch size, N is the sequence length, and d_{model} is the embedding dimension. The computational complexity of the position-oriented embedding is $O(B \cdot N \cdot d_{model})$, omitting the constant associated with the input length. The Gaussian-weighted sequence decomposition is essentially a 1D convolution operation applied along the last dimension d_{model}. Therefore, the computational complexity of this operation is $O(B \cdot N \cdot d_{model})$. **For the trend component**, it undergoes bidirectional Mamba-based representation learning, with a hidden dimension h_{dim}. The dimensional transitions follow $d_{model} \rightarrow h_{dim} \rightarrow d_{model}$, resulting in a computational complexity of $O(2 \cdot B \cdot N \cdot d_{model} \cdot h_{dim})$. **The fluctuation component** is processed using a TCN-based representation module, which involves three convolutional operations with kernel sizes k_1, k_2 and k_3, respectively. The dimensional transitions follow $d_{model} \rightarrow d_{model} \rightarrow d_{ff} \rightarrow d_{model}$, where d_{ff} is typically set to four times of d_{model}, resulting in a computational complexity of $O(B \cdot N \cdot d_{model} \cdot k_1 \cdot d_{model}) + O(B \cdot N \cdot d_{model} \cdot k_2 \cdot d_{ff}) + O(B \cdot N \cdot d_{ff} \cdot k_3 \cdot d_{model})$. Then, we perform an element-wise summation of the two components, which



((a)) NPO-SST ((b)) INO-SST

Figure 4: The results of STDMamba on the NPO-SST and INO-SST datasets while varying parameter kernel size, where the prediction length is 360.

has a complexity of $O(B \cdot N \cdot d_{model})$. Finally, we apply a linear projection to map the d_{model} dimension to the target prediction length, resulting in a computational complexity of $O(B \cdot N \cdot d_{model})$, where the constant term associated with the prediction length is omitted. In sum, the overall complexity is $O(4 \cdot B \cdot N \cdot d_{model} + 2 \cdot B \cdot N \cdot d_{model} \cdot h_{dim} + B \cdot N \cdot d_{model} \cdot k_1 \cdot d_{model} + B \cdot N \cdot d_{model} \cdot k_2 \cdot d_{ff} + B \cdot N \cdot d_{ff} \cdot k_3 \cdot d_{model})$. After simplifying the constant terms, the final complexity is $O((h_{dim} + d_{model} + d_{ff}) \cdot d_{model} \cdot N)$.

As shown in Table 1, Reformer, Informer, DLinear, iTransformer, and TimeMachine exhibit relatively high prediction efficiency. Reformer and Informer employ simple architectures with limited predictive capacity, whereas DLinear is a purely linear model; consequently, all three require few parameters and deliver good prediction efficiency. TimeMachine, which utilizes pure Mamba blocks, and iTransformer, which incorporates inverted embedding, achieve high prediction efficiency due to their architectural simplicity. ModernTCN, despite achieving fast inference, has a comparatively large number of parameters owing to its use of multiple stacked deep TCN blocks.

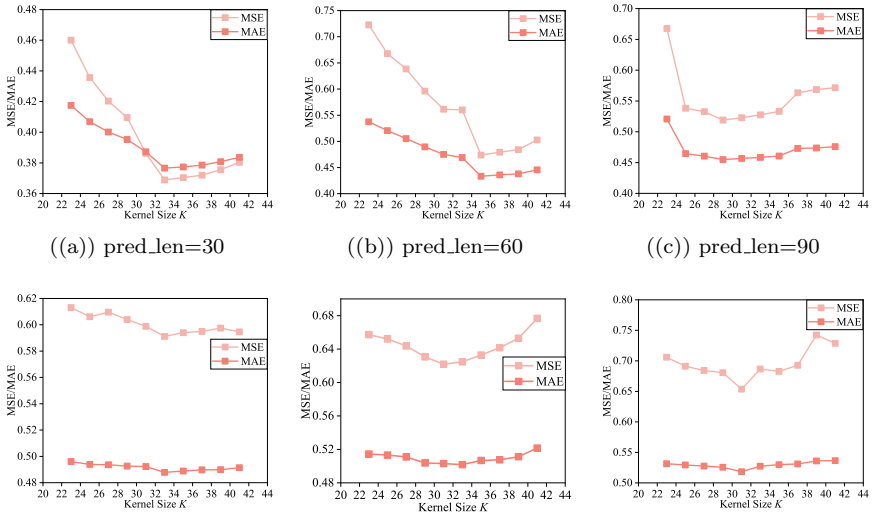
In contrast, our proposed STDMamba maintains a moderate level of efficiency in terms of parameter count, training time, and inference time. Although there is a slight increase in these metrics, it is primarily due to the incorporation of a decomposition module and a dual spatio-temporal representation learning mechanism, both of which significantly improve the model's ability to capture long-term fine-grained spatio-temporal dependencies, thereby enhancing prediction accuracy by a large margin.

Table 1: The prediction efficiency of STDMamba and baseline methods on the INO-SST dataset with input length $I = 360$ and prediction length $O = 360$.

Prediction methods	**Params Count(MB)**	**Train for 1 Iteration(s)**	**Infer for 1 Batch(ms)**	**MSE**
Reformer	2.15	0.21	14.04	0.6685
Informer	3.79	0.22	19.88	0.6262
DLinear	0.26	0.19	3.54	0.6052
iTransformer	3.36	0.29	8.28	0.6180
ModernTCN	458.03	0.42	9.23	0.6460
TimeMachine	17.44	0.39	7.00	0.6579
STDMamba	125.55	0.28	24.53	0.5629

C Parameter setting for STDMamba with Moviang Average method in the ablation experiment

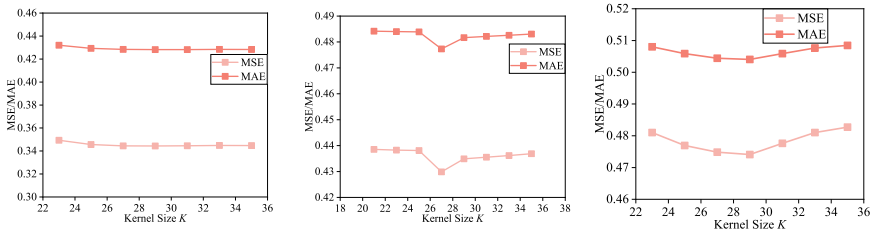
To ensure fair comparison, the optimal kernel size used for the moving average approach(MA) is also selected via experiments. Figures 5 and 6 present the results of STDMamba with MA for different prediction lengths while varying the kernel size.



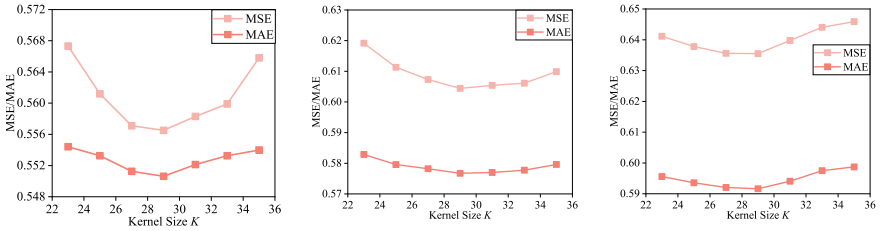
((a)) pred_len=30 ((b)) pred_len=60 ((c)) pred_len=90

((d)) pred_len=180 ((e)) pred_len=270 ((f)) pred_len=360

Figure 5: The prediction results of STDMamba with MA on the NPO-SST dataset for different prediction lengths (pred_len).



((a)) pred_len=30 ((b)) pred_len=60 ((c)) pred_len=90



((d)) pred_len=180 ((e)) pred_len=270 ((f)) pred_len=360

Figure 6: The prediction results of STDMamba with MA on the INO-SST dataset for different prediction lengths (pred_len).