SwinLSTM: Improving Spatiotemporal Prediction Accuracy using Swin Transformer and LSTM

Song Tang* Chuang Li* Pu Zhang RongNian Tang†
Hainan University
{songtang,lc,zhangpu,rn.tang}@hainanu.edu.cn

Abstract

Integrating CNNs and RNNs to capture spatiotemporal dependencies is a prevalent strategy for spatiotemporal prediction tasks. However, the property of CNNs to learn local spatial information decreases their efficiency in capturing spatiotemporal dependencies, thereby limiting their prediction accuracy. In this paper, we propose a new recurrent cell, SwinLSTM, which integrates Swin Transformer blocks and the simplified LSTM, an extension that replaces the convolutional structure in ConvLSTM with the self-attention mechanism. Furthermore, we construct a network with SwinLSTM cell as the core for spatiotemporal prediction. Without using unique tricks, SwinLSTM outperforms state-of-the-art methods on Moving MNIST, Human3.6m, TaxiBJ, and KTH datasets. In particular, it exhibits a significant improvement in prediction accuracy compared to ConvLSTM. Our competitive experimental results demonstrate that learning global spatial dependencies is more advantageous for models to capture spatiotemporal dependencies. We hope that SwinLSTM can serve as a solid baseline to promote the advancement of spatiotemporal prediction accuracy. The codes are publicly available at https://github.com/SongTang-x/SwinLSTM.

1. Introduction

Spatiotemporal prediction has received increasing attention in recent years due to it can benefit many practical applications, e.g., precipitation forecasting [24, 25, 33], autonomous driving [1,16], and traffic flow prediction [38,40]. However, the complex physical dynamics and chaotic properties of spatiotemporal predictive learning make it challenging for purely data-driven deep learning methods to make accurate predictions. Existing methods [3, 8, 19, 24, 31–34,39] integrate CNNs and RNNs to learn spatiotemporal dependencies in spatiotemporal data to improve prediction accuracy. To capture the spatial and temporal dependencies simultaneously, ConvLSTM [24] extends the fully connected LSTM (FC-LSTM) [26] by replacing linear operations with convolutional operations. Subsequently, several variants of ConvLSTM are proposed. PredRNN [33] and MIM [34] modify the internal structure of the LSTM unit. E3D-LSTM [32] integrates 3D-Convs into LSTMs. PhyD-Net [8] leverages a CNN-based module to disentangle physical dynamics. Admittedly, these methods achieve impressive results on spatiotemporal prediction tasks. However, convolution operators focus on capturing local features and relations and are inefficient for modeling global spatial information [4]. Although the receptive field can be enlarged by stacking convolution layers, the effective receptive field only reaches a fraction of the theoretical receptive field in practice [21]. Therefore, it can be inferred that the CNN-based models may prove to be ineffective in capturing spa-
Spatiotemporal dependencies, owing to the inherent locality of the CNN architecture, leading to restricted accuracy in predictions.

Recently, the Vision Transformer (ViT) [6] introduced the Transformer [29] that directly models long-range dependencies into the vision domain. ViT applies a standard Transformer for image classification and attains excellent results with sufficient data. Its outstanding achievement has attracted more researchers to apply Transformer to computer vision. Subsequently, some variants [20, 27, 36, 43] of ViT emerged, with tremendous success on different vision tasks. Notably, the Swin Transformer [20] has exhibited exceptional performance across diverse visual tasks, owing to its unique utilization of local attention and shift window mechanism.

Inspired by this, we propose SwinLSTM, a new recurrent cell. Specifically, we integrate Swin Transformer blocks and a simplified LSTM module to extract spatiotemporal representations. In addition, we construct a predictive network with SwinLSTM as the core to capture spatial and temporal dependencies for spatiotemporal prediction tasks. As illustrated in Figure 2 (c), we first split an input image at the current time step into a sequence of image patches. Subsequently, the flattened image patches are fed to the patch embedding layer. Then, the SwinLSTM layer receives the transformed patches or the hidden states transformed by the previous layer (Patch Merging or Patch Expanding), and the cell and hidden states of the previous time step to extract the spatiotemporal representations. Finally, the reconstruction layer decodes the spatiotemporal representations to generate the next frame.

The contributions of this paper can be summarized as follows:

- We propose a new recurrent cell, named SwinLSTM (section 3.2), which is able to efficiently extract spatiotemporal representations.
- We introduce a new architecture (section 3.1) for spatiotemporal prediction tasks, which can efficiently model spatial and temporal dependencies.
- We evaluate the effectiveness of the proposed model on Moving MNIST, TaxiBJ, Human3.6m, and KTH. Experimental results show that SwinLSTM achieves excellent performance on four datasets.

2. Related Work

CNN-Based Models  Prior models integrating CNNs and RNNs employ various strategies to better capture spatiotemporal dependencies to improve prediction accuracy. ConvLSTM [24] extends FC-LSTM [26] by replacing fully connected operations with convolutional operations to learn spatiotemporal dependencies. PredRNN [33] proposes the Spatiotemporal LSTM (ST-LSTM) module, which simultaneously models spatiotemporal information by transferring hidden states in horizontal and vertical directions. PredRNN++ [31] designs a Gradient Highway unit to solve the vanishing gradient problem in PredRNN. E3D-LSTM [32] replaces the 2D convolution in the ST-LSTM module with 3D convolution, allowing the ST-LSTM module to remember more previous information and achieve better prediction performance. MIM [34] replaces the forget gate in the ST-LSTM module with two recurrent units to solve the non-stationary information learning in prediction. CrevNet [39] proposes a CNN-based reversible network to learn complex spatiotemporal dependencies. PhyDNet [8] introduces physical knowledge into a CNN-based model to improve prediction quality. The above models [8, 24, 31–34, 39] improve their ability to capture spatiotemporal dependencies from different perspectives and achieve excellent results. However, despite their widespread usage, convolutional methods have inherent limitations in capturing spatiotemporal dependencies due to their local nature. In order to overcome this challenge, we propose the adoption of a global modeling approach (specifically, Swin Transformer blocks) for learning spatial dependencies, thereby enhancing the model’s capacity to capture spatiotemporal dependencies and improve prediction performance.

Vision Transformers The widespread use of Transformer [29] in the field of natural language processing (NLP) led researchers to introduce it to the vision domain. ViT [6] pioneered the direct application of the Transformer architecture to image classification and achieved great success. However, its excellent performance is based on training on large datasets, which leads to its unsatisfactory performance on smaller datasets. To solve this problem, DeiT [27] proposes several training strategies to allow ViT to perform well on the smaller ImageNet-1K [5] dataset. Subsequently, some variants [28, 37, 41, 42] of ViT achieved impressive results on various vision tasks. In particular, Swin Transformer [20] achieves outstanding performance on image classification, semantic segmentation, and object detection tasks due to its shifted window mechanism and hierarchical design. In this paper, we try to integrate the Swin Transformer blocks and the simplified LSTM to form a SwinLSTM recurrent cell, and use it as the core to build a model to capture temporal and spatial dependencies to perform spatiotemporal prediction tasks.

3. Method

3.1. Overall Architecture

The overall predictive architecture is depicted in Figure 2 (b and c). We build our base model and a deeper model, called SwinLSTM-B and SwinLSTM-D, respectively. First,
an image at the time step $t$ is split into non-overlapping patches. The patch size is $P^2$, and $P$ equals 2 or 4 in our implementation. Therefore, the feature dimension of each patch is $C \cdot P^2$ ($C$ denotes the number of channels). Subsequently, the image patches are flattened and fed to the patch embedding layer, which linearly maps the original features of the patches to an arbitrary dimension. Then, for SwinLSTM-B, the SwinLSTM layer receives the transformed image patches, the hidden state $H_{t-1}$, and cell state $C_{t-1}$ to generate the hidden state $H_t$ and cell state $C_t$, where the $H_t$ is duplicated into two copies, one for the reconstruction layer, the other together with $C_t$ for the SwinLSTM layer at the next time step. For SwinLSTM-D, we increase the number of SwinLSTM Cells, as well as add Patch Merging and Patch Expanding [2] layers, where the former is utilized for downsampling and the latter for upsampling. $H_t^{l=m}$ and $C_t^{l=m}$ refer to the hidden state and cell state at time step $t$ from layer $m$. Finally, the reconstruction layer maps the hidden state $H_t$ from the SwinLSTM layer to the input size to get the predicted frame for the next time step.
the Warm-up phase, we take the frames of the input sequence as input to the model. However, in the Prediction phase, we use the output of the model at the previous time step as the input to the model at the current time step. In particular, we concatenate the outputs of the Warm-up and Prediction phases to compute the loss. When the prediction loss function $L^p$ is $L_2$, its formula is shown in eq. 1.

$$L^p = ||[\hat{X}_{1:S:1}; \hat{I}_{1:S}] - [X_{1:S:1}; Y_{0:S:1}]||_2^2$$  \hspace{1cm} (1)

When the prediction loss function $L^p$ is $L_1 + L_2$, its formula is shown in eq. 2.

$$L^p = ||[\hat{X}_{1:S:1}; \hat{I}_{1:S}] - [X_{1:S:1}; Y_{0:S:1}]||_1$$
$$+ ||[\hat{X}_{1:S:1}; \hat{I}_{1:S}] - [X_{1:S:1}; Y_{0:S:1}]||_2^2$$  \hspace{1cm} (2)

In eq. 1 and eq. 2, $\hat{X}_{1:S:1}$ represents S-1 predicted frames in the Warm-up phase, $\hat{I}_{1:S}$ denotes S predicted frames in the Prediction phase, and $X_{1:S:1}$ and $Y_{0:S:1}$ represent S-1 input frames and S ground truth future frames, respectively.

### 3.2. SwinLSTM Module

ConvLSTM [24] introduces the convolution operators into input-to-state and state-to-state transitions to overcome the shortcomings of FC-LSTM [26] in processing spatiotemporal data. The key equations of ConvLSTM are as follows:

$$i_t = \sigma (W_{xi} \ast x_t + W_{hi} \ast h_{t-1} + b_i)$$
$$f_t = \sigma (W_{xf} \ast x_t + W_{hf} \ast h_{t-1} + b_f)$$
$$o_t = \sigma (W_{xo} \ast x_t + W_{ho} \ast h_{t-1} + b_o)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh (W_{xc} \ast x_t + W_{hc} \ast h_{t-1} + b_c)$$
$$h_t = o_t \odot \tanh (C_t)$$  \hspace{1cm} (3)

where `*` denotes the convolution operator and `;' denotes the Hadamard product. $i_t$, $f_t$, $o_t$ are input, forget, output gate. Different from convolution operators to extract local correlations, the self-attention mechanism captures global spatial dependencies by computing similarity scores across all positions. Thus, we remove all weights $W$ and biases $b$ in eq. 3 to obtain eq. 4:

$$i_t = f_t = o_t = \sigma (x_t + h_{t-1})$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh (x_t + h_{t-1})$$
$$h_t = o_t \odot \tanh (C_t)$$  \hspace{1cm} (4)

Obviously, $i_t = f_t = o_t$ in eq. 4. Therefore, we fuse the three gates: $i_t$, $f_t$, $o_t$ into one gate, named filter gate $F_t$.

The detailed structure of the SwinLSTM Module is presented in Figure 2 (a). In SwinLSTM, the information of cell states $C_t$ and hidden states $H_t$ is updated horizontally to capture long-term and short-term temporal dependencies. Meanwhile, the Swin Transformer blocks vertically learns global spatial dependencies. The key equations of SwinLSTM are shown in eq. 5, where STB denotes the Swin Transformer blocks and LP denotes the Linear Projection:

$$F_t = \sigma (STB (LP (X_t; H_{t-1})))$$
$$C_t = F_t \odot (\tanh (STB (LP (X_t; H_{t-1}))) + C_{t-1})$$
$$H_t = F_t \odot \tanh (C_t)$$  \hspace{1cm} (5)

### 3.3. Swin Transformer Block

The global multi-head self-attention (MSA) mechanism in the Vision Transformer [6] computes the relationship between a token and all other tokens. Such a computation leads to quadratic computational complexity related to the number of tokens, which is not friendly to dense prediction tasks. To alleviate the computational burden of global self-attention and improve the modeling power, Swin Transformer [20] proposes window-based multi-head self-attention (W-MSA) and shifted-window-based multi-head self-attention (SW-MSA).

$$\hat{z}^l = W \ast \text{MSA} (\text{LN} (z^{l-1})) + z^{l-1}$$
$$z^l = \text{MLP} (\text{LN} (\hat{z}^l)) + \hat{z}^l$$
$$z^{l+1} = W \ast \text{MSA} (\text{LN} (z^l)) + z^l$$
$$z^{l+1} = \text{MLP} (\text{LN} (z^{l+1})) + \hat{z}^{l+1}$$  \hspace{1cm} (6)
### Table 1. Experimental setup. SwinLSTMs denotes the number of the SwinLSTM Cells in predictive network. STB denotes the number of the Swin Transformer blocks in SwinLSTM cell. Patch size indicates the patch token size. Train and Test represent the number of input and output frames during training and testing.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>SwinLSTMs</th>
<th>STB</th>
<th>Patch size</th>
<th>Resolution</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving MNIST</td>
<td>SwinLSTM-D</td>
<td>4</td>
<td></td>
<td>(2, 6, 6, 2)</td>
<td>(64, 64, 1)</td>
<td>10 → 10</td>
<td>10 → 10</td>
</tr>
<tr>
<td>Human3.6m</td>
<td>SwinLSTM-B</td>
<td>1</td>
<td>12</td>
<td>(128, 128, 3)</td>
<td>(64, 64, 1)</td>
<td>4 → 4</td>
<td>4 → 4</td>
</tr>
<tr>
<td>KTH</td>
<td>SwinLSTM-B</td>
<td>1</td>
<td></td>
<td>(128, 128, 1)</td>
<td>(64, 64, 1)</td>
<td>10 → 10</td>
<td>10 → 20/40</td>
</tr>
<tr>
<td>TaxiBJ</td>
<td>SwinLSTM-B</td>
<td>1</td>
<td>12</td>
<td>(32, 32, 2)</td>
<td>(64, 64, 1)</td>
<td>4 → 4</td>
<td>4 → 4</td>
</tr>
</tbody>
</table>

As shown in Figure 3, the first Swin Transformer block is based on W-MSA, consisting of two LayerNorm layers and a 2-layer MLP with GELU non-linearity. A LayerNorm layer and a residual connection are applied before and after each W-MSA module and MLP module, respectively. The second Swin Transformer block is the same as the first except that the W-MSA block in it is replaced by an SW-MSA block. The key equations of Swin Transformer blocks are shown in eq. 6.

### 4. Experiments

We evaluate our proposed model on four commonly used datasets: Moving MNIST [26], TaxiBJ [40], Human3.6m [10], and KTH [23], and present quantitative comparison results and visual examples for each dataset, illustrating the effectiveness and generalizability of our proposed neural network. In addition, we perform ablation studies (section 4.5) and feature map visualization (section 4.6).

#### 4.1. Implementations

We use $L_2$ loss for the MovingMNIST dataset, $L_1 + L_2$ loss for the Human3.6m, the TaxiBJ, and the KTH datasets. The input and target frames of each dataset are normalized to the intensity of $[0, 1]$. Each model is optimized with the Adam optimizer [15]. We train our model on an Nvidia RTX A5000 GPU. More detailed parameters for each dataset are listed in Table 1.

#### 4.2. Evaluation Metrics

We adopt the Mean Squared Error (MSE), the Mean Absolute Error (MAE), the Peak Signal to Noise Ratio (PSNR), and the Structural Similarity Index Measure (SSIM) [35] as metrics to evaluate the quality of predictions. All metrics are averaged over predicted frames. Lower MAE and MSE or higher SSIM and PSNR indicate better prediction accuracy.

#### 4.3. Datasets

**Moving MNIST.** The moving MNIST dataset [26] is a widely used synthetic dataset. In our experiments, the method for generating moving MNIST sequences follows [26], each sequence contains 20 frames, and the first 10 frames and the last 10 frames are used for input and target, respectively. The two handwritten digits in each frame are randomly sampled in the MNIST dataset [17], moving and bouncing around a $64 \times 64$ pixel black canvas with a fixed speed and angle. We can get an infinite number of se-
Figure 4. Qualitative results of SwinLSTM on four datasets. The first line is the input, the second line is the ground truth, and the third line is the prediction of SwinLSTM. For Moving MNIST, we add the prediction results of ConvLSTM as the fourth line. For TaxiBJ, we add the absolute difference between the predictions and ground truths as the fourth line. For KTH action, (c) is the jogging action when the model predicts 20 frames based on the previous 10 frames, and (e) is the walking action when the model observes 10 frames and predicts the next 40 frames.

quences by applying different speeds and angles to different handwritten digits. We adopt 10000 sequences for training and a fixed set containing 10000 sequences for testing.

**Human3.6m.** The human3.6m dataset [10] contains 3.6 million different human poses and corresponding images. Following [34], we only use the ‘walking’ scenario. Subjects S1, S5, S6, S7, S8 are used for training, and S9, S11 are used for testing. The images in the dataset are resized from $1000 \times 1000 \times 3$ to $128 \times 128 \times 3$. We train the models to predict the next 4 RGB frames from 4 observations.

**KTH.** The KTH dataset [23] contains 25 individuals performing 6 categories of human actions (walking, jogging, running, boxing, hand waving, and hand clapping) in 4 different scenarios. We follow the experimental setup in [30], resizing each image to $128 \times 128$ and using persons 1-16 for training and 17-25 for testing. The models predict 10
frames from 10 observations at training time and 20 or 40 frames at inference time.

**TaxiBJ.** TaxiBJ [40] contains complex real-world taxi trajectory data collected from taxi cab GPS monitors in Beijing. Each frame in TaxiBJ is a $32 \times 32 \times 2$ heat map, where the last dimension represents the flow of traffic entering and leaving the same area. We follow the experimental setup in [40] and use the last four weeks of data for testing and the rest for training. We use 4 observations to predict the next 4 consecutive frames. We adopted per-frame MSE as the rest for training. We use 4 observations to predict the trajectory data collected from taxicab GPS monitors in Beijing. We follow the experimental setup in [40] and use the last four weeks of data for testing and the rest for training. We use 4 observations to predict the trajectory data collected from taxicab GPS monitors in Beijing. We follow the experimental setup in [40] and use the last four weeks of data for testing and the rest for training. We use 4 observations to predict the next 4 consecutive frames. We adopted per-frame MSE as metrics.

### 4.4. Main results

Table 2, 3, 4 and 5 present quantitative comparisons with previous state-of-the-art models on four datasets. On Moving MNIST, SwinLSTM outperforms previous CNN-based models and achieves the new state-of-the-art results. Specifically, compared to ConvLSTM [24], SwinLSTM reduces the MSE from 103.3 to 17.7 and increases the SSIM from 0.707 to 0.962. On both Human3.6m and KTH, SwinLSTM achieves the state-of-the-art, especially with huge gains of 4.49 (10→20) and 5.59 (10→40) in PSNR on KTH. In the case of TaxiBJ, our proposed SwinLSTM achieved a remarkable reduction in MSE for each predicted frame compared to previous models. Additionally, the inter-frame MSE difference was smaller than that of the compared models. The results indicate SwinLSTM’s great potential for an efficient and generalizable technique for spatiotemporal prediction.

In Figure 4, we present qualitative results of SwinLSTM on all datasets. (a) presents a qualitative analysis of SwinLSTM and ConvLSTM on an extremely hard case of Moving MNIST. In the presented images, two digits are continuously intertwined, posing a challenge to the model’s ability to accurately predict their motion trajectories and appearances. As demonstrated, both models effectively capture the trajectory of the digits. However, in terms of appearance, ConvLSTM’s prediction results become increasingly blurred as the number of time steps increases, while SwinLSTM’s predictions maintain high similarity throughout. SwinLSTM, as an extension of ConvLSTM, utilizes LSTM [9] to capture temporal dependencies, and employs Swin Transformer blocks [20] to model global spatial information. In contrast, ConvLSTM uses convolution to model local spatial information. The significant performance gap between the two models demonstrates that learning global spatial information can better assist the model in capturing spatiotemporal dependencies and thus improve prediction accuracy.

To facilitate the observation of the similarity between the predicted results and the ground truth in (b), we visualize the absolute difference between them. For (c), (d), and (e), we observe that SwinLSTM makes accurate predictions of human actions and scenes, where (c) and (e) show visual examples of 2 different categories of actions predicted by SwinLSTM, Jogging and walking respectively. The model receives 10 frames during training and predicts 10 frames, whereas it predicts 20 or 40 frames during testing. Expanding the predicted number of frames during testing poses a significant challenge to the model, requiring it to effectively learn spatiotemporal dependencies from the spatiotemporal data. Our experiments on the KTH dataset demonstrate that SwinLSTM outperforms alternative models in capturing spatiotemporal dependencies.

### 4.5. Ablation Study

In this section, we perform ablation studies on TaxiBJ and Human3.6m to analyze the impact of different elements on model performance. Specifically, we discuss three major elements: the reconstruction layer, patch size, and the number of Swin Transformer blocks.

**The reconstruction layer.** The role of the reconstruction layer is to decode the spatiotemporal representations extracted by the SwinLSTM cell. We conduct experiments on transposed convolution, bilinear interpolation, and lin-
ear projection to analyze the impact of different decoding methods. Table 6 shows that the transposed convolution performs much better than the other two methods.

<table>
<thead>
<tr>
<th>The reconstruction layer</th>
<th>TaxiBJ MSE</th>
<th>SSIM</th>
<th>Human3.6m MSE</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transposed convolution</td>
<td>0.390</td>
<td>0.980</td>
<td>33.2</td>
<td>0.913</td>
</tr>
<tr>
<td>Bilinear interpolation</td>
<td>0.794</td>
<td>0.961</td>
<td>39.3</td>
<td>0.887</td>
</tr>
<tr>
<td>Linear Projection</td>
<td>0.415</td>
<td>0.979</td>
<td>72.5</td>
<td>0.773</td>
</tr>
</tbody>
</table>

Table 6. Ablation study on the three different methods of reconstruction layer with SwinLSTM on TaxiBJ and Human3.6m.

Patch size. The size of the image patch directly determines the length of the input token sequence. The smaller the size, the longer the length. To explore the effect of different patch sizes on prediction accuracy, we conduct experiments with patch sizes set to 2, 4, and 8 on TaxiBJ and Human3.6m. Table 7 shows that the optimal patch size varies across datasets, which indicates that adjusting the patch size to an appropriate size is beneficial to improve the generalization of SwinLSTM.

<table>
<thead>
<tr>
<th>Patch size</th>
<th>TaxiBJ MSE</th>
<th>SSIM</th>
<th>Human3.6m MSE</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.417</td>
<td>0.979</td>
<td>33.2</td>
<td>0.913</td>
</tr>
<tr>
<td>4</td>
<td>0.390</td>
<td>0.980</td>
<td>40.2</td>
<td>0.885</td>
</tr>
<tr>
<td>8</td>
<td>0.393</td>
<td>0.980</td>
<td>51.9</td>
<td>0.839</td>
</tr>
</tbody>
</table>

Table 7. Ablation study on different patch sizes on TaxiBJ and Human3.6m.

The number of Swin Transformer Blocks. Based on the mechanism of Swin Transformer blocks (STB) modeling global spatial information, it is evident that the learning capacity of the model varies with the number of STB utilized. We explored the effect of the number of STB from 2 to 18 on model performance. Figure 5 presents the results of different numbers of STB on TaxiBJ and Human3.6m.

4.6. Feature map Visualization

To analyze the mechanism of the SwinLSTM cell (section 3.2), we randomly selected a sample from the Moving MNIST test set and visualized the feature maps during inference. Figure 6 shows the results, where white arrows have been added to the images in the third row to facilitate the observation of digit trajectories. The Hidden States capture the position of digits at the current time step, whereas the Cell States memorize the digit trajectory. Meanwhile, we conducted a visualization of the feature maps generated by the second, fourth, and sixth Swin Transformer blocks of the SwinLSTM-B. By observing the feature maps of STB-2 to STB-6, it is evident that STB gradually learns global spatial correlations as the number of interactions of window self-attention increases (section 3.3). By analyzing the visualizations generated by our model, we aimed to gain valuable insights into its performance and behavior. These insights will inform future optimizations and improvements.
5. Conclusion

In this paper, we propose SwinLSTM, a new recurrent cell that incorporates Swin Transformer blocks and simplified LSTM, and constructs a predictive network for spatiotemporal prediction tasks based on it. Through extensive experimentation, it has been demonstrated that the proposed method, SwinLSTM, achieves excellent performance on various datasets, including Moving MNIST, TaxiBJ, Human3.6m, and KTH. These impressive results showcase the effectiveness and generalization ability of SwinLSTM. Notably, SwinLSTM outperforms its predecessor, ConvLSTM, in prediction accuracy, which suggests that learning global spatial dependencies allows the model to better capture spatiotemporal dependencies. Overall, this study highlights the potential of SwinLSTM as a promising approach for spatiotemporal modeling.

Acknowledgement

This work was supported by the Natural Science Foundation Innovation Research Team Project of Hainan Province (320CXTD431), the National Key Research and Development Program of China (2021YFB1507104), the National Natural Science Foundation of China (32060413), and the Major Science and Technology Project of Hainan Province (ZDKJ2020013), which are gratefully acknowledged.

References


[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer:


