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.

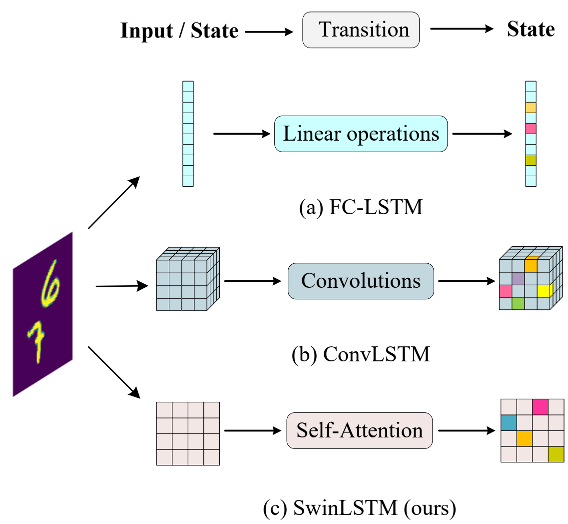


Figure 1. Comparison of input-to-state and state-to-state transitions in three different recurrent cells. (a) FC-LSTM utilizes linear operations to process 1D vectors. (b) ConvLSTM employs 2D convolutions to process 3D tensors. (c) The proposed SwinLSTM leverages self-attention to process 2D matrices.

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. PhyDNet [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-

* Equal contribution. †Corresponding author.

tiotemporal 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,

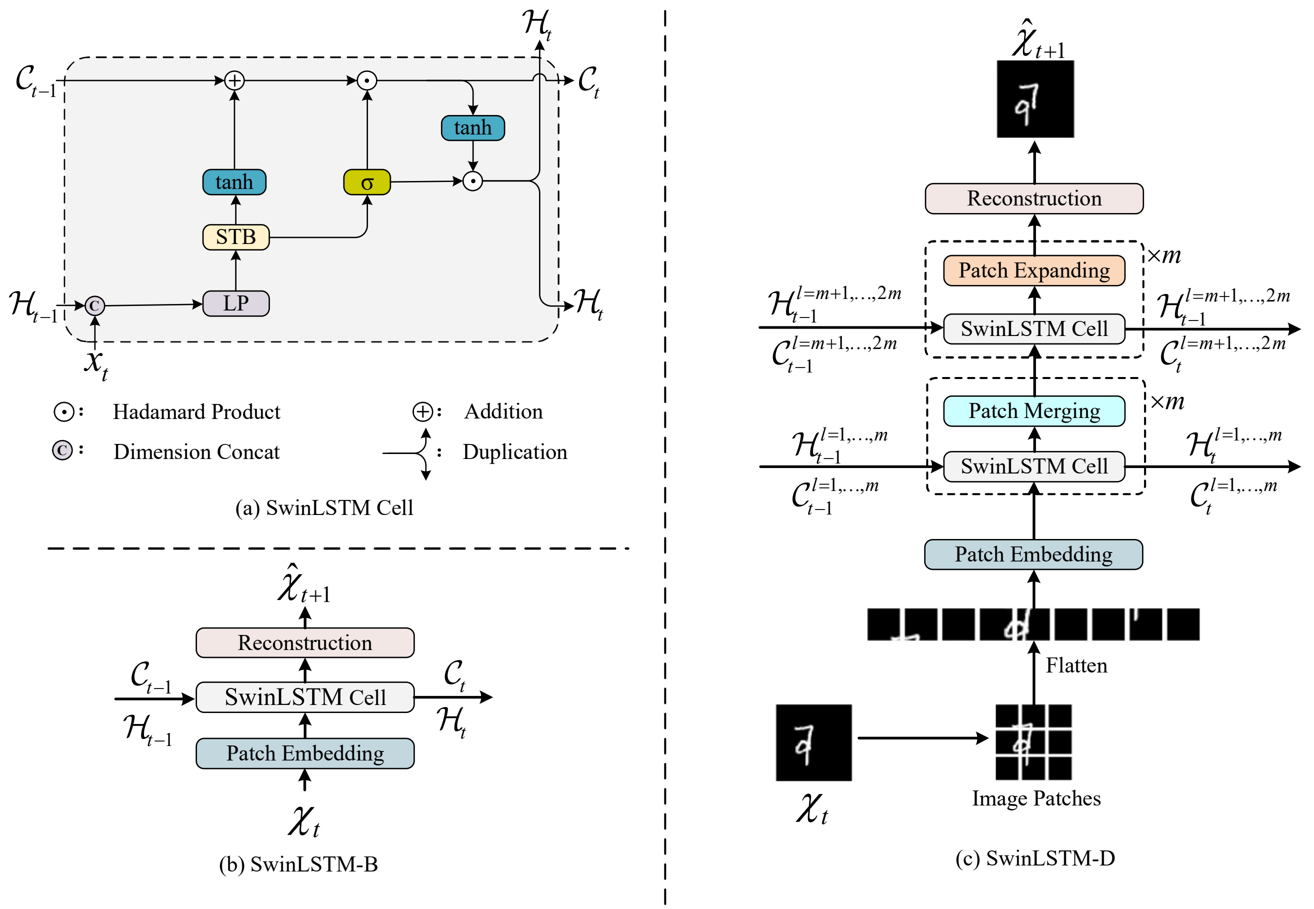


Figure 2. (a): The detailed structure of the proposed recurrent cell: SwinLSTM. **STB** and **LP** denote Swin Transformer blocks and Linear Projection. (b): The architecture of the base model with a single SwinLSTM cell. (c): The architecture of the deep model with multiple SwinLSTM cells.

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 $\mathcal{H}_{t-1}$, and cell state $\mathcal{C}_{t-1}$ to generate the hidden state $\mathcal{H}_t$ and cell state $\mathcal{C}_t$, where the $\mathcal{H}_t$ is duplicated into two copies, one for the reconstruction layer, the other together with $\mathcal{C}_t$ 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. $\mathcal{H}_t^{l=m}$ and $\mathcal{C}_t^{l=m}$ refer to the hidden state and cell state at time step t from layer m. Finally, the reconstruction layer maps the hidden state $\mathcal{H}_t$ from the SwinLSTM layer to the input size to get the predicted frame for the next time step.

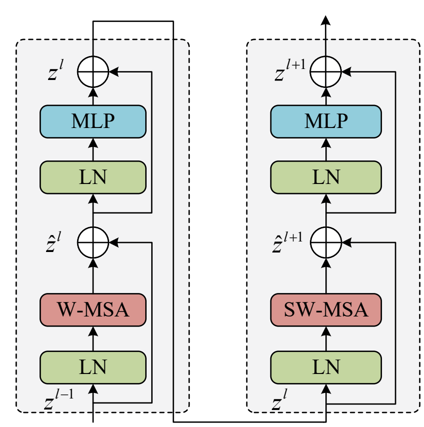


Figure 3. Swin Transformer Block.

Both SwinLSTM-B and SwinLSTM-D were trained using the same training approach. The specific training process for SwinLSTM-B is presented in Algorithm 1. During

Algorithm 1 Training pipeline

Require: α, the learning rate. m, the batch size. n, the number of training epochs. S, the number of frames in the input (target) sequence.
Initialize parameters of network (θ).
for $i = 0, 1, ..., n\text{-}1$ **do**
  for number of batches **do**
    Sample $\{[X_{0:S\text{-}1}; Y_{0:S\text{-}1}]^{(j)}\}_{j=1}^{m}$ a batch from the training set.
    Phase 1: Warm-up Phase
    Initialize $\mathcal{H}_0, \mathcal{C}_0$
    for $t = 0, 1, ..., S\text{-}2$ **do**
      Get $\hat{X}_{t+1}, \mathcal{H}_{t+1}, \mathcal{C}_{t+1} = P(X_t, \mathcal{H}_t, \mathcal{C}_t)$
    end for
    Phase 2: Prediction Phase
    $\hat{I}_0 \leftarrow X_{S\text{-}1}, \mathcal{H}_0 \leftarrow \mathcal{H}_{S\text{-}1}, \mathcal{C}_0 \leftarrow \mathcal{C}_{S\text{-}1}$
    for $t = 0, 1, ..., S\text{-}1$ **do**
      Get $\hat{I}_{t+1}, \mathcal{H}_{t+1}, \mathcal{C}_{t+1} = P(\hat{I}_t, \mathcal{H}_t, \mathcal{C}_t)$
    end for
    $\mathcal{L} \leftarrow \mathcal{L}^{\text{P}}([\hat{X}_{1:S\text{-}1}; \hat{I}_{1:S}], [X_{1:S\text{-}1}; Y_{0:S\text{-}1}])$
    Update $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$
  end for
end for

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 $\mathcal{L}^{\text{P}}$ is L_2, its formula is shown in eq. 1.

$$\mathcal{L}^{\text{P}} = \|[\hat{X}_{1:S\text{-}1}; \hat{I}_{1:S}] - [X_{1:S\text{-}1}; Y_{0:S\text{-}1}]\|_2^2 \tag{1}$$

When the prediction loss function $\mathcal{L}^{\text{P}}$ is $L_1 + L_2$, its formula is shown in eq. 2.

$$\begin{aligned}\mathcal{L}^{\text{P}} =& \|[\hat{X}_{1:S\text{-}1}; \hat{I}_{1:S}] - [X_{1:S\text{-}1}; Y_{0:S\text{-}1}]\|_1 \\ &+ \|[\hat{X}_{1:S\text{-}1}; \hat{I}_{1:S}] - [X_{1:S\text{-}1}; Y_{0:S\text{-}1}]\|_2^2\end{aligned} \tag{2}$$

In eq. 1 and eq. 2, $\hat{X}_{1:S\text{-}1}$ represents S-1 predicted frames in the **Warm-up** phase, $\hat{I}_{1:S}$ denotes S predicted frames in the **Prediction** phase, and $X_{1:S\text{-}1}$ and $Y_{0:S\text{-}1}$ 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:

$$\begin{aligned}
i_t &= \sigma\left(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + b_f\right) \\
o_t &= \sigma\left(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + b_o\right) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\left(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c\right) \\
\mathcal{H}_t &= o_t \circ \tanh\left(\mathcal{C}_t\right)
\end{aligned} \tag{3}$$

where '*' denotes the convolution operator and '$\circ$' 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:

$$\begin{aligned}
i_t &= f_t = o_t = \sigma\left(\mathcal{X}_t + \mathcal{H}_{t-1}\right) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\left(\mathcal{X}_t + \mathcal{H}_{t-1}\right) \\
\mathcal{H}_t &= o_t \circ \tanh\left(\mathcal{C}_t\right)
\end{aligned} \tag{4}$$

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 $\mathcal{F}_t$.

The detailed structure of the SwinLSTM Module is presented in Figure 2 (a). In SwinLSTM, the information of cell states $\mathcal{C}_t$ and hidden states $\mathcal{H}_t$ 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:

$$\begin{aligned}
\mathcal{F}_t &= \sigma\left(\text{STB}\left(\text{LP}\left(\mathcal{X}_t; \mathcal{H}_{t-1}\right)\right)\right) \\
\mathcal{C}_t &= \mathcal{F}_t \circ \left(\tanh\left(\text{STB}\left(\text{LP}\left(\mathcal{X}_t; \mathcal{H}_{t-1}\right)\right)\right) + \mathcal{C}_{t-1}\right) \\
\mathcal{H}_t &= \mathcal{F}_t \circ \tanh\left(\mathcal{C}_t\right)
\end{aligned} \tag{5}$$

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).

$$\begin{aligned}
\hat{\mathbf{z}}^l &= \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1} \\
\mathbf{z}^l &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^l\right)\right) + \hat{\mathbf{z}}^l \\
\hat{\mathbf{z}}^{l+1} &= \text{SW-MSA}\left(\text{LN}\left(\mathbf{z}^l\right)\right) + \mathbf{z}^l \\
\mathbf{z}^{l+1} &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1}
\end{aligned} \tag{6}$$

Dataset	Model	SwinLSTMs	STB	Patch size	Resolution	Train	Test
Moving MNIST	SwinLSTM-D	4	(2, 6, 6, 2)	2	(64, 64, 1)	$10 \rightarrow 10$	$10 \rightarrow 10$
Human3.6m	SwinLSTM-B	1	12	2	(128, 128, 3)	$4 \rightarrow 4$	$4 \rightarrow 4$
KTH	SwinLSTM-B	1	6	4	(128, 128, 1)	$10 \rightarrow 10$	$10 \rightarrow 20/40$
TaxiBJ	SwinLSTM-B	1	12	4	(32, 32, 2)	$4 \rightarrow 4$	$4 \rightarrow 4$

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.

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

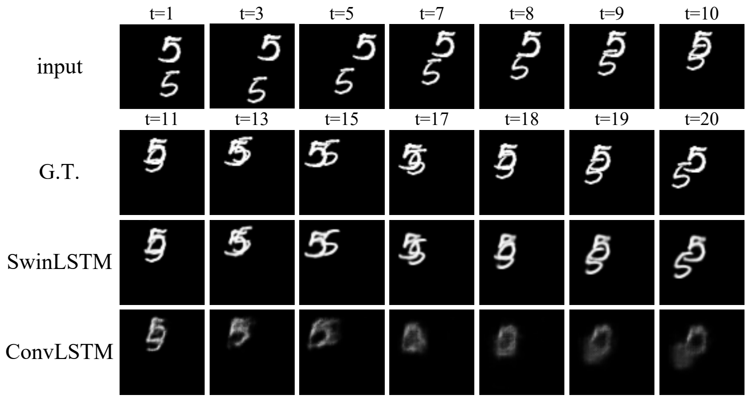
Moving MNIST. The moving MNIST dataset [26] is a widely used synthetic dataset. In our experiments, the method for generating moving MNIST sequences follows [26], each sequence contains 20 frames, and the first 10 frames and the last 10 frames are used for input and target, respectively. The two handwritten digits in each frame are randomly sampled in the MNIST dataset [17], moving and bouncing around a 64×64 pixel black canva with a fixed speed and angle. We can get an infinite number of se-

Method	MSE $\downarrow$	SSIM $\uparrow$
ConvLSTM [24]	103.3	0.707
DFN [11]	89.0	0.726
FRNN [22]	69.7	0.813
VPN [14]	64.1	0.870
PredRNN [33]	56.8	0.867
CausalLSTM [31]	46.5	0.898
MIM [34]	44.2	0.910
E3D-LSTM [32]	41.3	0.910
LMC [19]	41.5	0.924
MAU [3]	27.6	0.937
PhyDNet [8]	24.4	0.947
CrevNet [39]	22.3	0.949
SwinLSTM	**17.7**	**0.962**

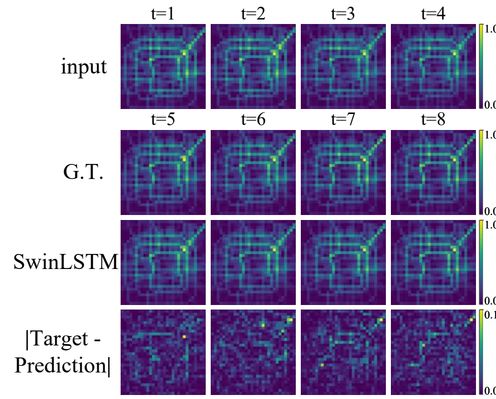
Table 2. Quantitative comparison of SwinLSTM and other methods on **Moving MNIST**. Each model observes 10 frames and predicts the subsequent 10 frames. Lower MSE and higher SSIM indicate better predictions.

Method	MAE /100 $\downarrow$	MSE /10 $\downarrow$	SSIM $\uparrow$
FRNN [22]	19.0	49.8	0.771
ConvLSTM [24]	18.9	50.4	0.776
PredRNN [33]	18.9	48.4	0.781
CausalLSTM [31]	17.2	45.8	0.851
MIM [34]	17.8	42.9	0.790
E3D-LSTM [32]	16.6	46.4	0.869
PhyDNet [8]	16.2	36.9	0.901
SwinLSTM	**11.9**	**33.2**	**0.913**

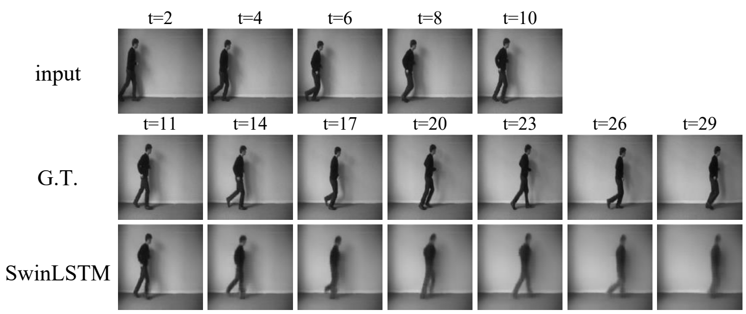
Table 3. Quantitative evaluation of different models on the **Human3.6m** dataset.



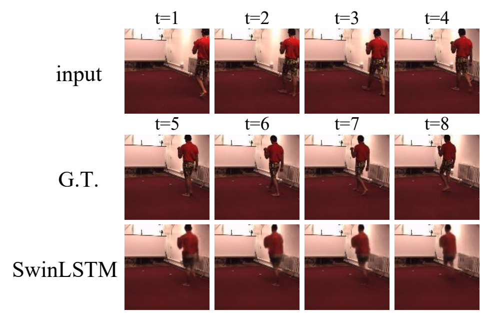
(a) Moving MNIST



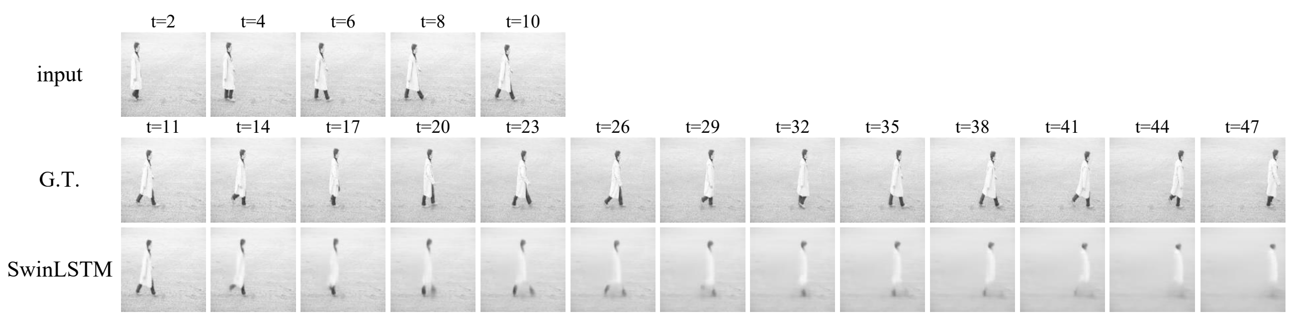
(b) TaxiBJ



(c) KTH (Jogging 10→20)



(d) Human3.6m



(e) KTH (Walking 10→40)

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 128×128 and using persons 1-16 for training and 17-25 for testing. The models predict 10

Method	KTH (10 → 20)		KTH (10 → 40)	
	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑
ConvLSTM [24]	0.712	23.58	0.639	22.85
SAVP [7]	0.746	25.38	0.701	23.97
FRNN [22]	0.771	26.12	0.678	23.77
DFN [11]	0.794	27.26	0.652	23.01
PredRNN [33]	0.839	27.55	0.703	24.16
VarNet [13]	0.843	28.48	0.739	25.37
SVAP-VAE [18]	0.852	27.77	0.811	26.18
PredRNN++ [31]	0.865	28.47	0.741	25.21
E3d-LSTM [32]	0.879	29.31	0.810	27.24
STMFANet [12]	0.893	29.85	0.851	27.56
SwinLSTM	**0.903**	**34.34**	**0.879**	**33.15**

Table 4. Quantitative evaluation on the **KTH** test set. We present the model observing 10 frames to predict 20 or 40 frames, and all metrics are averaged over the predicted frames. Higher SSIM and PSNR indicate better prediction quality.

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 taxicab 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 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.

Method	Frame1 ↓	Frame2 ↓	Frame3 ↓	Frame4 ↓
VPN [14]	0.744	1.031	1.251	1.444
ST-ResNet [40]	0.688	0.939	1.130	1.288
FRNN [22]	0.682	0.823	0.989	1.183
PredRNN [33]	0.634	0.934	1.047	1.263
PredRNN++ [31]	0.641	0.855	0.979	1.158
E3d-LSTM [32]	0.620	0.773	0.888	1.158
MIM [34]	0.554	0.737	0.887	0.999
SwinLSTM	**0.324**	**0.401**	**0.473**	**0.525**

Table 5. Quantitative results on the **TaxiBJ** dataset. We report the per-frame MSE of the 4 predictions.

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 linear projection to analyze the impact of different decoding methods. Table 6 shows that the transposed convolution performs much better than the other two methods.

The reconstruction layer	TaxiBJ MSE	TaxiBJ SSIM	Human3.6m MSE /10	Human3.6m SSIM
Transposed convolution	**0.390**	**0.980**	**33.2**	**0.913**
Bilinear interpolation	0.794	0.961	39.3	0.887
Linear Projection	0.415	0.979	72.5	0.773

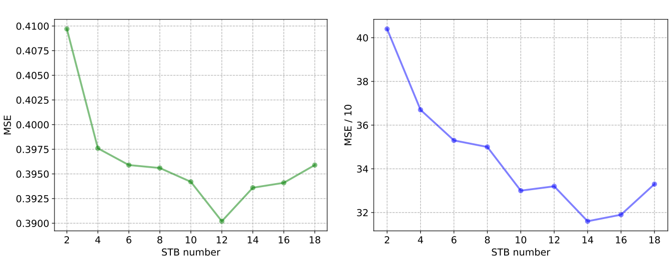
Table 6. Ablation study on the three different methods of reconstruction layer with SwinLSTM on TaxiBJ and Human3.6m.

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.

Patch size	TaxiBJ MSE	TaxiBJ SSIM	Human3.6m MSE /10	Human3.6m SSIM
2	0.417	0.979	**33.2**	**0.913**
4	**0.390**	**0.980**	40.2	0.885
8	0.393	0.980	51.9	0.839

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 present the results of different numbers of STB on TaxiBJ and Human3.6m.



(a) TaxiBJ (b) Human3.6m

Figure 5. Ablation study on the different numbers of STB with SwinLSTM on TaxiBJ and Human3.6m.

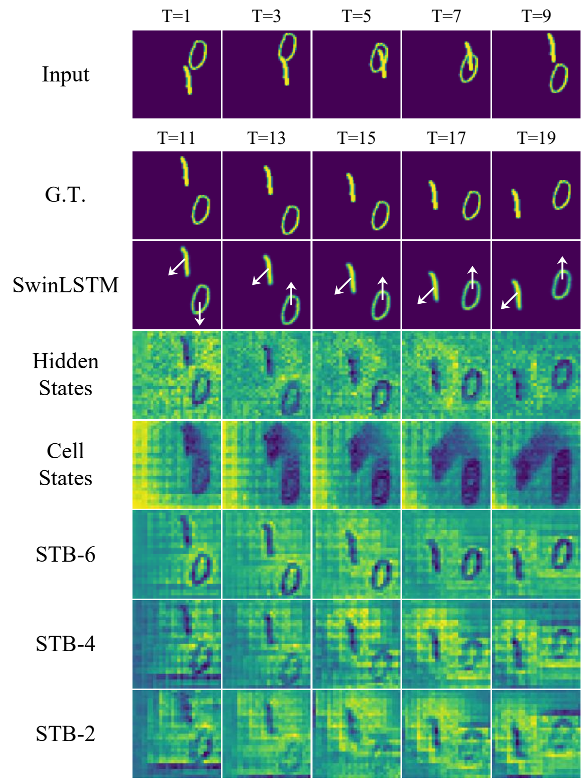


Figure 6. Feature map visualization of a sample in the Moving MNIST test set. **Hidden States** and **Cell States** are the feature maps of $\mathcal{H}_t$ and $\mathcal{C}_t$ of the recurrent cell SwinLSTM (section 3.2) at different time steps. **STB-2**,**STB-4**, and **STB-6** are the feature maps of the 2nd, 4th, and 6th layers of the Swin Transformer blocks (section 3.3), respectively.

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

[1] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4194–4202, 2018. 1

[2] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 205–218. Springer, 2023. 3

[3] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. Mau: A motion-aware unit for video prediction and beyond. *Advances in Neural Information Processing Systems*, 34:26950–26962, 2021. 1, 5

[4] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. *Advances in neural information processing systems*, 31, 2018. 1

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 4

[7] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9006–9015, 2019. 7

[8] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020. 1, 2, 5

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 7

[10] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 5, 6

[11] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016. 5, 7

[12] Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4554–4563, 2020. 7

[13] Beibei Jin, Yu Hu, Yiming Zeng, Qiankun Tang, Shice Liu, and Jing Ye. Varnet: Exploring variations for unsupervised video prediction. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5801–5806. IEEE, 2018. 7

[14] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning*, pages 1771–1779. PMLR, 2017. 5, 7

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 5

[16] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019. 1

[17] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 5

[18] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 7

[19] Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. Video prediction recalling long-term motion context via memory alignment learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3054–3063, 2021. 1, 5

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

Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 4, 7
[21] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016. 1
[22] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 716–731, 2018. 5, 7
[23] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004. 5, 6
[24] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 1, 2, 4, 5, 7
[25] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in neural information processing systems*, 30, 2017. 1
[26] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015. 1, 2, 4, 5
[27] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2
[28] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 2
[29] 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. 2
[30] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017. 6
[31] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, pages 5123–5132. PMLR, 2018. 1, 2, 5, 7
[32] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International conference on learning representations*, 2018. 1, 2, 5, 7
[33] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017. 1, 2, 5, 7
[34] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9154–9162, 2019. 1, 2, 5, 6, 7
[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5
[36] Yu-Huan Wu, Yun Liu, Xin Zhan, and Ming-Ming Cheng. P2t: Pyramid pooling transformer for scene understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2
[37] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 2
[38] Ziru Xu, Yunbo Wang, Mingsheng Long, Jianmin Wang, and M KLiss. Predcnn: Predictive learning with cascade convolutions. In *IJCAI*, pages 2940–2947, 2018. 1
[39] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. 2020. 1, 2, 5
[40] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 1, 5, 7
[41] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3008, 2021. 2
[42] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 2
[43] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 2