Self-Attention ConvLSTM for Spatiotemporal Prediction

Zhihui Lin,[1,2] **Maomao Li,**[2] **Zhuobin Zheng,**[1,2] **Yangyang Cheng,**[1,2] **Chun Yuan**[2,3]*
[1]Department of Computer Science and Technologies, Tsinghua University, Beijing, China
[2]Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[3]Peng Cheng Laboratory, Shenzhen, China
{lin-zh14, mm-li17, zhengzb16, cheng-yy13}@mails.tsinghua.edu.cn, yuanc@sz.tsinghua.edu.cn

Abstract

Spatiotemporal prediction is challenging due to the complex dynamic motion and appearance changes. Existing work concentrates on embedding additional cells into the standard ConvLSTM to memorize spatial appearances during the prediction. These models always rely on the convolution layers to capture the spatial dependence, which are local and inefficient. However, long-range spatial dependencies are significant for spatial applications. To extract spatial features with both global and local dependencies, we introduce the self-attention mechanism into ConvLSTM. Specifically, a novel self-attention memory (SAM) is proposed to memorize features with long-range dependencies in terms of spatial and temporal domains. Based on the self-attention, SAM can produce features by aggregating features across all positions of both the input itself and memory features with pair-wise similarity scores. Moreover, the additional memory is updated by a gating mechanism on aggregated features and an established highway with the memory of the previous time step. Therefore, through SAM, we can extract features with long-range spatiotemporal dependencies. Furthermore, we embed the SAM into a standard ConvLSTM to construct a self-attention ConvLSTM (SA-ConvLSTM) for the spatiotemporal prediction. In experiments, we apply the SA-ConvLSTM to perform frame prediction on the MovingMNIST and KTH datasets and traffic flow prediction on the TexiBJ dataset. Our SA-ConvLSTM achieves state-of-the-art results on both datasets with fewer parameters and higher time efficiency than previous state-of-the-art method.

1 Introduction

Spatiotemporal predictive learning has emerged as an important and foundational research problem for a wide range of computer vision and artificial intelligence and received growing interests in the research communities (Shi et al. 2015; Zhang et al. 2017; Shi et al. 2017a; Kalchbrenner et al. 2017; Wang et al. 2017a; 2018b; Xu et al. 2018; Wang et al. 2019). It deserves to be studied in depth to serve the practical applications, such as traffic flows prediction (Zhang et al. 2017; Xu et al. 2018), precipitation forecasting (Shi et al. 2015; 2017b; Wang et al. 2017b) and physical interactions simulation (Lerer, Gross, and Fergus 2016; Finn, Goodfellow, and Levine 2016). Spatiotemporal prediction is challenging due to the complex dynamics and appearance changes, which requires dependencies on both temporal and spatial domains.

ConvLSTM (Shi et al. 2015) replaces all the linear operations in it with convolution layers to capture spatial dependencies besides the long-short term modeling, and many of its variants (Wang et al. 2017a; 2018b; 2019) have achieved impressive results on spatiotemporal prediction. However, although long-range spatial dependencies can be captured by stacked convolution layers, the effective receptive field is much smaller than the theoretical receptive field (Luo et al. 2016). Besides, features far away from a specific location have to pass through a stack of layers before affecting the location for both forward propagation and backward propagation, which would add the optimization difficulties during the training (Chen et al. 2018). Therefore, ConvLSTM and its previous variants tend to suffer from the limited ability to capture long-range spatial dependencies. To ameliorate this, TrajGRU (Shi et al. 2017b) adopts a convolution layer to learn offsets of each position in the hidden state of a GRU block. It works in a similar way to the deformable convolution (Dai et al. 2017), which enhances it in modeling complex object deformations. Nevertheless, these offsets only provide sparse spatial dependencies and are estimated with the local receptive field. Here comes to a question that how to make the ConvLSTM capture effective long-range dependencies.

Compared to the convolution operation, the self-attention module (Vaswani et al. 2017; Wang et al. 2018a) is capable of obtaining the global spatial context with a single layer, which is more efficient. Besides, we argue that features at the current time step can benefit from aggregating relevant features in the past. Therefore, we propose the self-attention memory module for ConvLSTM, or SAM in short. SAM utilizes the feature aggregation mechanism of the self-attention to fuse both the current and memorized features through calculating pair-wise similarity scores. Here, we use an additional memory cell $\mathcal{M}$ to memorize previous features which contains global spatial receptive field. Besides in the spa-

*Corresponding Author: Chun Yuan

tial domain, $\mathcal{M}$ can capture long-range temporal dependencies through a gating mechanism, which is similar to that in LSTM. The SAM is embedded into ConvLSTM to construct the self-attention ConvLSTM, or SA-ConvLSTM in short. We evaluate the above models on MovingMNIST and KTH for multi-frame prediction, and TexiBJ for traffic flow prediction. Ablation experiments demonstrate the effectiveness of self-attention and additional memory on different types of data. Moreover, SA-ConvLSTM achieves the best results on all datasets with fewer parameters and higher efficiency than previous state-of-the-art methods. Our contribution can be summarized as follows:

- We propose a novel variant of ConvLSTM, named SA-ConvLSTM to perform spatiotemporal prediction, which can successfully capture long-range spatial dependencies.
- We design a memory-based self-attention module (SAM) to memorize the global spatiotemporal dependencies during the prediction.
- We evaluate SA-ConvLSTM on MovingMNIST and KTH for multi-frame prediction and TexiBJ for traffic flow prediction. It achieves the best results in all datasets with fewer parameters and higher efficiency than the current state-of-the-art model MIM.

2 Related Work

Spatiotemporal Prediction with ConvRNNs. Variants of ConvLSTM (Shi et al. 2015) have been proposed to conduct spatiotemporal prediction. PredRNN (Wang et al. 2017b) introduced an additional spatiotemporal memory cell to propagate information across both horizontal and vertical directions with the highway connection, which is helpful to model spatial dynamics. PredRNN++ (Wang et al. 2018b) increases the transition depth by re-organizing the memories of PredRNN in a cascade fashion. To enhance the ability of PredRNN on modeling high-order dynamics, Memory in Memory (MIM) (Wang et al. 2019) introduces more memory cells to process non-stationary and stationary information, which achieves the current SOTA performance in spatiotemporal prediction while result in a multiplication of computation and memory usage. All of them stack convolution layers to obtain spatial dependences since deeper networks can be exponentially more efficient in capturing both spatial and temporal dependences (Bianchini and Scarselli 2014; Pascanu, Mikolov, and Bengio 2013). However, it is easy for them to suffer from the vanishing gradient problem (Bengio et al. 1994; Pascanu, Mikolov, and Bengio 2013). Besides, although the above additional cells for memorizing spatial appearance improve the model capacity of ConvLSTM models, their memory cells tend to focus on local spatial dependences. In this paper, we propose a self-attention memory cell for ConvLSTM, which can not only obtain the long-term temporal dependence through the adaptive updating in the highway but also efficiently extract the global spatial dependence through self-attention.

Self-Attention Modules. The self-attention mechanism is first proposed to draw global dependencies of inputs and applied in machine translation (Vaswani et al. 2017). As for computer vision tasks, self-attention is able to capture long-range spatial-temporal dependencies by calculating the pairwise relations among the different position of feature maps during a binary relation function. Then the attended features can be calculated through these relations (Zhang et al. 2019). Then, several variants (Fu et al. 2019; Chen et al. 2018; Huang et al. 2019) were proposed for more efficient computing or more diverse attention types. The successes of self-attention on pixel-level tasks (Huang et al. 2019; Fu et al. 2019; Zhang et al. 2019) demonstrate its effectiveness on aggregating salient features among all spatial positions. In this paper, We utilize the property of self-attention to construct a self-attention memory module and embed it into the ConvLSTM as SA-ConvLSTM, which is capable of bringing global dependency effectively.

3 Methods

In order to evaluate the effectiveness of self-attention in spatiotemporal prediction, we construct a basic self-attention ConvLSTM model by cascading self-attention module and the standard ConvLSTM, which is detailed in Section 3.1. Afterwards, a more advanced and sophisticated model SA-ConvLSTM is built based on the proposed self-attention memory module, which is introduced in Section 3.3.

3.1 Base Model

The base model is a simple combination of self-attention and ConvLSTM; that is, the base model is built by the direct cascade of two parts. This base model is formulated as follows:

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

where SA denotes the self-attention module. $\hat{\mathcal{X}}$ and $\hat{\mathcal{H}}$ are aggregated features through self-attention modules. Specifically, at each time step, the position at attention module selectively aggregates the input feature at each position by a weighted sum of the feature at all positions. This makes the global spatial dependencies can be captured during propagation cross stacked RNN layers vertically and through all RNN states horizontally. However, self-attention has very high computational complexity in high-resolution input since it needs to calculate the correlation among all positions. In this work, the size of images is small such that the complexity of SA can be ignored to a certain extent.

Self-Attention. Figure 1 shows the pipeline of the standard self-attention module. The original feature maps $\mathcal{H}_t$ are mapped into different feature spaces as the **query**: $\mathbf{Q}_h = \mathbf{W}_q\mathcal{H}_t \in \mathbb{R}^{\hat{C}\times N}$, the **key**: $\mathbf{K}_h = \mathbf{W}_k\mathcal{H}_t \in \mathbb{R}^{\hat{C}\times N}$ and the **value**: $\mathbf{V}_h = \mathbf{W}_v\mathcal{H}_t \in \mathbb{R}^{C\times N}$, where $\{\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v\}$ is a

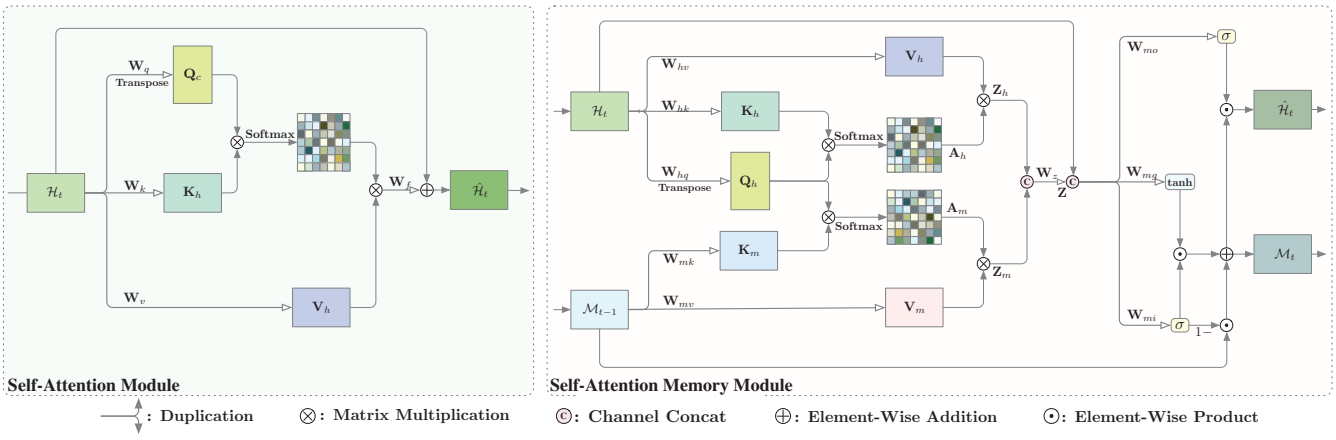

Figure 1: The illustration of the standard self-attention module and the proposed self-attention memory module, or SAM in short. In the self-attention module, $\mathcal{H}_t$ is the hidden state in ConvLSTM at the time step t, $\mathbf{Q}_h$ is the **query**, $\mathbf{K}_h$ indicates the **key**, $\mathbf{V}_h$ represents the **value** based on the 1×1 convolution on the feature, and $\hat{\mathcal{H}}_t$ is the output. As for the proposed SAM, the aggregated feature $\mathbf{Z}_h$ is obtained by applying self-attention on $\mathcal{H}_t$ and another feature $\mathbf{Z}_m$, where $\mathbf{Z}_m$ is calculated by querying on $\mathbf{K}_m$ and visiting $\mathbf{V}_m$. Here, both of $\mathbf{K}_m$ and $\mathbf{V}_m$ are mappings of the memory $\mathcal{M}_{t-1}$. $\mathbf{Z}_h$ and $\mathbf{Z}_m$ are fused as $\mathbf{Z}$ by 1×1 convolution. Then $\mathbf{Z}$ and original input $\mathcal{H}_t$ is used to update the memory with a gating mechanism. The final output is a dot product between the output gate value and the updated memory $\mathcal{M}_t$.

set of weights for 1×1 convolutions, C and $\hat{C}$ are number of channels, and where $N = H \times W$.

The similarity scores of each pair of points are calculated by applying the matrix production as:

$$\mathbf{e} = \mathbf{Q}_h^T \mathbf{K}_h \in \mathbb{R}^{N \times N}. \tag{2}$$

The similarity between the i-th point and the j-th point can be indexed as $e_{i,j} = (\mathcal{H}_{t,i}^T \mathbf{W}_q^T)(\mathbf{W}_k \mathcal{H}_{t,j})$ where the $\mathcal{H}_{t,i}$ and the $\mathcal{H}_{t,j}$ are feature vectors with the shape $C \times 1$. Then, the similarity scores are normalized along columns:

$$\alpha_{i,j} = \frac{\exp e_{i,j}}{\sum_{k=1}^{N} \exp e_{i,k}}, \; i, j \in \{1, 2, ..., N\}. \tag{3}$$

The aggregated feature of the i-th location is calculated with a weighted sum across all locations:

$$\mathbf{Z}_i = \sum_{j=1}^{N} \alpha_{i,j} (\mathbf{W}_v \mathcal{H}_{t;j}), \tag{4}$$

where $\mathbf{W}_v \mathcal{H}_{t,j} \in \mathbb{R}^{C \times 1}$ is the j-th column of the **value** $\mathbf{V_h}$. The output is obtained with a shortcut connection $\hat{\mathcal{H}}_t = \mathbf{W}_f \mathbf{Z} + \mathcal{H}_t$. Here, the residual mechanism stables the model training and ensures the module is flexible to be embedded into other deep models.

3.2 Self-Attention Memory Module

We argue that the prediction of the current time step can benefit from the past relevant features. Therefore, we propose a self-attention memory module by constructing a new-designed memory unit $\mathcal{M}$ with the self-attention mechanism. We use the proposed memory unit to represent the general spatiotemporal information which has the global spatial and temporal receptive field.

The structure of the proposed self-attention memory is illustrated in Figure 1. Our self-attention memory block receives two inputs, the input feature $\mathcal{H}_t$ at the current time step and the memory $\mathcal{M}_{t-1}$ at the last step. The whole pipeline can be separated into three parts, the **feature aggregation** to obtain the global context information, the **memory updating** and the **output**.

Feature Aggregation. At each time step, the aggregated feature $\mathbf{Z}$ is the fusion of $\mathbf{Z}_h$ and $\mathbf{Z}_m$. $\mathbf{Z}_h$ is acquired in the same way as the self-attention described in the section 3.1. $\mathbf{Z}_m$ is aggregated by querying on memory at the last time step $\mathcal{M}_{t-1}$. The memory is mapped into **key** $\mathbf{K_m} \in \mathbb{R}^{\hat{C} \times N}$ and **value** $\mathbf{V_m} \in \mathbb{R}^{C \times N}$ by 1×1 convolutions through weights $\mathbf{W}_{mk}$ and $\mathbf{W}_{mv}$. Then, similarity scores between the input and the memory are calculated by the matrix multiplication between the **query** $\mathbf{Q}_h$ and the **key** $\mathbf{K_m}$:

$$\mathbf{e}_m = \mathbf{Q}_h^T \mathbf{K_m} \in \mathbb{R}^{N \times N}. \tag{5}$$

Similar to Equation 3, all weights which are used for aggregating features are obtained by applying SoftMax function along each row, same as the Eq. 3:

$$\alpha_{m;i,j} = \frac{\exp e_{m;i,j}}{\sum_{k=1}^{N} \exp e_{m;i,k}}, \; i, j \in \{1, 2, ..., N\}. \tag{6}$$

Then, the 'pixel' of i-th location in feature $\mathbf{Z}_m$ is calculated by a weighted sum across all N locations in **value** $\mathbf{V}_m$.

$$\mathbf{Z}_{m;i} = \sum_{j=1}^{N} \alpha_{m;i,j} \mathbf{V}_{m;j} = \sum_{j=1}^{N} \alpha_{m;i,j} \mathbf{W}_{mv} \mathcal{M}_{t-1;j}, \tag{7}$$

where $\mathcal{M}_{t-1;j}$ is the j-the column of the memory. Finally, the aggregated feature $\mathbf{Z}$ can be obtained with $\mathbf{Z} = \mathbf{W}_z[\mathbf{Z}_h; \mathbf{Z}_m]$.

Memory Updating. We adopt a gating mechanism to update the memory $\mathcal{M}$ adaptively, such that the SAM can capture long-range dependencies in terms of spatial and temporal domains. The aggregated feature $\mathbf{Z}$ and the original input $\mathcal{H}_t$ are used to produce values of the input gate $i_t^{'}$ and the fused feature $g_t^{'}$. Besides, the forget gate is replaced as $1 - i_t^{'}$ to reduce parameters. The updating progress can be formulated as follows:

$$\begin{aligned} i_t^{'} &= \sigma(W_{m;zi} * \mathbf{Z} + W_{m;hi} * \mathcal{H}_t + b_{m;i}) \\ g_t^{'} &= \tanh(W_{m;zg} * \mathbf{Z} + W_{m;hg} * \mathcal{H}_t + b_{m;g}) \\ \mathcal{M}_t &= (1 - i_t^{'}) \circ \mathcal{M}_{t-1} + i_t^{'} \circ g_t^{'} \end{aligned} \tag{8}$$

Here, to further reduce parameters and computation, we replace the standard operation with depth-wise separable convolution (Chollet 2017). Compared with the original memory cell $\mathcal{C}$ in the ConvLSTM which is updated by convolution operations only, the proposed memory $\mathcal{M}$ is updated by not only convolution operations but also aggregated features $\mathcal{Z}_t$, obtaining the global spatial dependency timely. Therefore, we argue that $\mathcal{M}_{t-1}$ is able to contain global past spatiotemporal information.

Output. Finally, the output feature $\hat{\mathcal{H}}_t$ of the self-attention memory module is a dot product between the output gate $o_t^{'}$ and updated memory $\mathcal{M}_t$, which can be formulated as follows:

$$\begin{aligned} o_t^{'} &= \sigma(W_{m;zo} * \mathbf{Z} + W_{m;ho} * \mathcal{H}_t + b_{m;o}) \\ \hat{\mathcal{H}}_t &= o_t^{'} \circ \mathcal{M}_t \end{aligned} \tag{9}$$

3.3 Self-Attention ConvLSTM

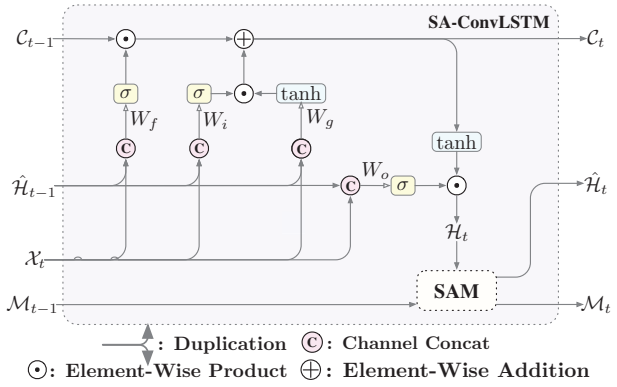

Figure 2: The self-attention ConvLSTM (SA-ConvLSTM) block. The **SAM** is the self-attention memory which is detailed in Figure 1.

We embed the self-attention memory module into the ConvLSTM to construct the SA-ConvLSTM, as illustrated in Figure 2. If we remove the **SAM** module, the SA-ConvLSTM will degenerate into the standard ConvLSTM. Besides, it is flexible to embedded in other models.

4 Experiments

We make the spatiotemporal prediction on three commonly used datasets, including the MovingMNIST and KTH for multi-frame prediction, and the TexiBJ for the traffic flow prediction. To demonstrate the effect of the proposed memory unit and the self-attention mechanisms, we first carry out an ablation study on the MovingMNIST and the TexiBJ, which is detailed in Section 4.3. Then, we show the quantitative results on each dataset in Section 4.4. We also provide the visualization examples to demonstrate the superiority of proposed SA-ConvLSTM on the spatiotemporal prediction. Moreover, to explain the effect of self-attention mechanism in the proposed SA-ConvLSTM, we visualize the attention maps from the first and last layers.

4.1 Implementation

To make fair comparisons with previous work (Shi et al. 2015; Wang et al. 2017b; 2018b; 2019; Xu et al. 2018), we apply almost the same experiment setting, that is, a 4-layer architecture with 64 hidden states in each layer for every model. The scheduled sampling strategy (Bengio et al. 2015) and LayerNorm (Ba, Kiros, and Hinton 2016) are also adopted in the training process. Each model is trained with an ADAM optimizer and a beginning learning rate of 0.001. During training, the mini-batch is set to 8, and the training process is stopped after 80,000 iterations. We use $L2$ loss for the MovingMNIST and the TaxiBJ datasets, while $L1 + L2$ loss for the KTH dataset.

4.2 Datasets

MovingMNIST is a commonly used dataset contains a variety of sequences generated by the method mentioned in (Srivastava et al. 2015), depicting two potentially overlapping digits moving with constant velocity and bouncing off the image edges. Image size is $64 \times 64 \times 1$, and each sequence contains 20 frames with 10 inputs and 10 for prediction.

TaxiBJ is collected from the chaotic real-world environment and contains traffic flow images collected consecutively from the GPS monitors of taxicabs in Beijing. Each frame in TaxiBJ is a $32 \times 32 \times 2$ grid of image. Two channels represent the traffic flow entering and leaving the same district at this time. We use 4 known frames to predict the next 4 frames (traffic conditions for the next two hours).

KTH (Schuldt et al. 2004) contains 6 categories of human actions, including boxing, hand waving, hand clapping, walking, jogging and running, completed by 25 people in 4 different scenarios. We follow the setup in previous works (Oliu et al. 2018; Zhang et al. 2016; Wang et al. 2017a; 2018b; 2019) to construct the training and testing sets. Image size are resized from 320×240 to 128×128. 10 frames are used to predict the next 10 frames during training and 20 frames at inference.

4.3 Ablation Study

We perform an ablation study on the MovingMNIST and the TexiBJ to evaluate models on the different types of data. The motion change in the MovingMNIST is smooth; therefore, performing predictions on this dataset requires accurate modeling of local dynamics. In contrast, the TexiBJ uses the evolution of pixel value to represent traffic flow variation. Thus, the TexiBJ has more long-range spatial dependencies than the MovingMNIST.

Table 1: Ablation study on the MovingMNIST and the TexiBJ datasets. We use SSIM, MSE, MAE to measure the prediction quality. ConvLSTM is the baseline model, and four variants are evaluated, including the base model in Section 3.1, ConvLSTM with additional memory, and SA-ConvLSTM with or without $\mathbf{Z}_m$ in Figure 1.

Datasets	MovingMNIST						TexiBJ					
Models	SSIM↑	Δ	MSE↓	Δ	MAE↓	Δ	SSIM↑	Δ	MSE↓	Δ	MAE↓	Δ
ConvLSTM	0.852	–	63.98	–	133.34	–	0.979	–	0.527	–	4.253	–
w SA, w/o Mem	0.869	+0.017	58.25	-5.73	118.08	-15.26	0.982	+0.003	0.410	-0.117	3.881	-0.372
w/o SA, w Mem	0.872	+0.020	56.17	-7.81	114.02	-19.32	0.982	+0.003	0.431	-0.096	3.948	-0.305
w SA, w Mem, w/o $\mathbf{Z}_m$	0.884	+0.032	55.60	-8.38	113.19	-20.15	0.982	+0.003	0.408	-0.119	3.872	-0.381
w SA, w Mem, w $\mathbf{Z}_m$	**0.913**	**+0.061**	**43.92**	**-20.06**	**94.73**	**-38.61**	**0.984**	**+0.005**	**0.390**	**-0.137**	**3.822**	**-0.431**

To verify the effectiveness of the self-attention and the additional memory $\mathcal{M}$, we apply five different models, including 1) the standard 4-layer ConvLSTM, 2) the base model which is constructed as in Figure 2 with self-attention, 3) the ConvLSTM with additional memory cell $\mathcal{M}$ but without the self-attention part, and 4) the SA-ConvLSTM without Z_m in Figure 1, 5) the complete SA-ConvLSTM, as illustrated as in Figure 2. We Adopt the SSIM (structural similarity Index Measure) (Wang et al. 2004), MSE (Mean Square Error) and MAE (Mean Absolute Error) as metrics, where MSE and MAE measure the pixel-level differences, which are more suitable for synthetic data.

Experimental results are shown in Table 1. Self-attention relatively reduces MSE by 9.0% and 22.2% on MovingMNIST and TexiBJ separately. As for the additional memory $\mathcal{M}$, the relative reductions are 12.2% and 18.2%. Additional memory is more effective on data with smooth dynamics, while self-attention is more suitable for traffic or network flow prediction since it can extract long-range spatial dependencies. SA-ConvLSTM (w/o $\mathbf{Z}_m$) achieves MSE reductions by 13.1% and 22.6% on the MovingMNIST and the TexiBJ separately. The whole SA-ConvLSTM combines both the advantages, which reduces MSE by 32.2% and 26.0% on these two types of data. Aggregating past features from the additional memory with global spatial and temporal dependencies is very crucial for SA-ConvLSTM.

4.4 Quantitative and Qualitative Comparison

MovingMNIST. Quantitative comparisons among different models are detailed in Table 2, where the averaged results are reported. We apply the PredRNN (Wang et al. 2017a), PredRNN++ (Wang et al. 2018b), MIM (Wang et al. 2019) and other models as the comparison, where MIM achieves the state-of-the-art methods in recent years. All models predict the next 10 frames based on 10 previous frames. We follow the experiment settings and hyper-parameters of the PredRNN, PredRNN++, and MIM for a fair comparison.

Compared to the PredRNN, our base model has fewer parameters and achieves comparable results, which shows the self-attention boost ConvLSTM to a large extent. Our SA-ConvLSTM has a smaller model scale than PredRNN or PredRNN++. The parameters of SA-ConvLSTM are even less than half of that in the SOTA model MIM or MIM*, where MIM* is based on the CausalLSTM (Wang et al. 2018b), instead of the ConvLSTM. The smaller model scale is due to the adoption of depth-wise separable convolution (Chollet 2017) in the proposed self-attention memory, reducing the trainable parameters. All of the PredRNN, PredRNN++, and MIM rely on convolutions to extract spatial dependencies, which is limited and inefficient. In contrast, SA-ConvLSTM achieves the best results on all measurements. In particular, our model reduces the MAE by 6.4 than MIM*, obtaining more accurate and sharper predictions. We also evaluated the efficiency of each model based on a GTX 1080TI GPU and the TensorFlow framework. ConvLSTM costs 0.42s for one forward-backward iteration, PredRNN spends 0.66s, MIM costs 1.14s, while PredRNN++ and MIM* take longer. Our base model costs 0.54s and SA-ConvLSTM costs 0.72s,

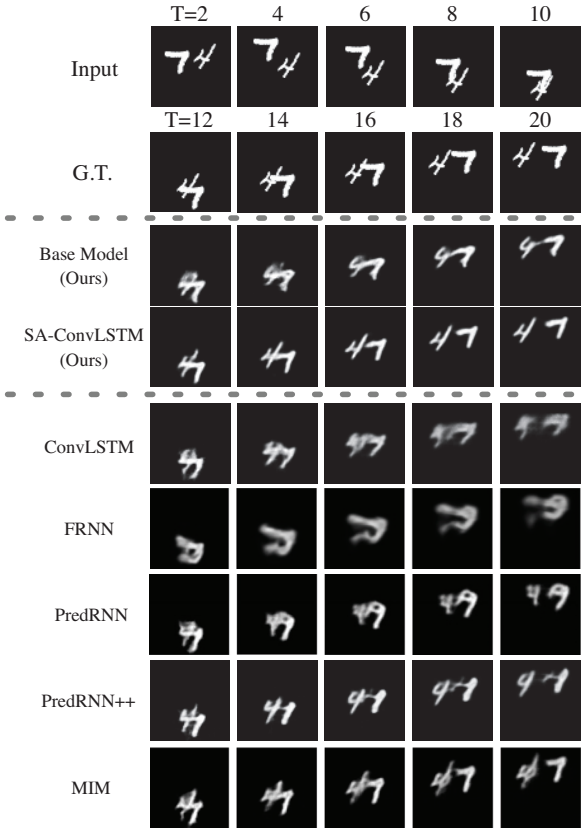

Figure 3: Qualitative comparison of different models on the MovingMNIST test set. All models predict 10 frames into the future by observing 10 previous frames.

Table 2: Qualitative Comparison of different models on the MovingMNIST. All models predict 10 frames into the future by observing 10 previous frames. The output frames are shown at two-frame intervals.

Models	#Params	SSIM↑	Δ	MSE↓	Δ	MAE↓	Δ
FC-LSTM (Srivastava et al. 2015)	–	0.690	–	118.3	–	209.4	–
ConvLSTM (Shi et al. 2015)	–	0.707	+0.017	103.3	-15.0	182.9	-26.5
TrajGRU (Shi et al. 2017b)	–	0.713	+0.023	106.9	-11.4	190.1	-19.3
DFN (Jia et al. 2016)	–	0.726	+0.036	89.0	-28.3	172.8	-36.6
FRNN (Oliu et al. 2018)	–	0.813	+0.123	69.7	-48.6	150.3	-59.1
VPN baseline (Kalchbrenner et al. 2017)	–	0.870	+0.180	64.1	-54.2	131.0	-78.4
PredRNN (Wang et al. 2017a)	13.799M	0.867	+0.177	56.8	-61.5	126.1	-83.3
MIM (Wang et al. 2019)	28.533M	0.874	+0.184	52.0	-66.3	116.5	-92.9
PredRNN++ (Wang et al. 2018b)	13.237M	0.898	+0.208	46.5	-71.8	106.8	-102.6
MIM*	27.971M	**0.910**	**+0.220**	**44.2**	**-74.1**	**101.1**	**-108.3**
Base Model (Ours)	10.102M	0.869	+0.179	58.3	-60.0	118.1	-91.3
SA-ConvLSTM (Ours)	10.471M	**0.913**	**+0.223**	**43.9**	**-74.4**	**94.7**	**-114.7**

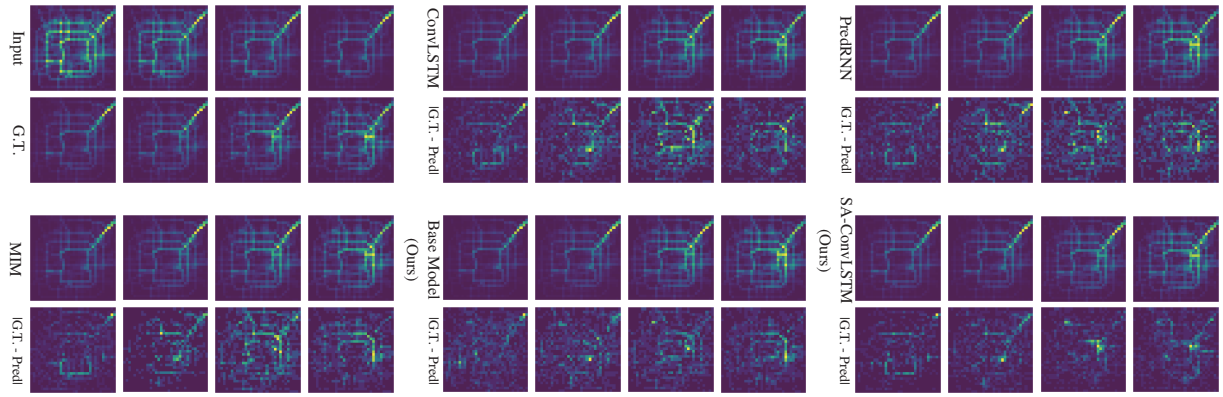

Figure 4: Visualization samples of on the TaxiBJ test set. All models output next 4 frames conditioned on the last 4 frames. The absolute differences between predictions and ground truths are shown. The brighter the color, the higher of the absolute errors.

which is around 37% faster than MIM.

The qualitative comparison of each model is visualized in Figure 3. The FRNN (Oliu et al. 2018) and ConvLSTM produce vaguest results. Results of PredRNN, PredRNN++, and MIM are still too blurry to distinguish the digits '4' and '7'. Our base model achieves sharp but not very precise predictions. SA-ConvLSTM achieves the best predictions in terms of accuracy and image quality.

TaxiBJ. Quantitative comparisons on the TaxiBJ test set is detailed in Table 3a. Each model predicts the next 4 frames (traffic conditions for the next two hours) by 4 known frames. We adopt the frame-wise MSE as the metric. The visualized comparisons are shown in Figure 4, which includes both frame and the absolute difference between prediction results and the ground truth frame. Besides, the proposed SA-ConvLSTM reduces the averaged MSE error by around 9.3% than the MIM.

KTH. Table 3b shows quantitative comparisons among previous state-of-the-art methods and SA-ConvLSTM on the KTH dataset. We use 10 last frames to predict the next 20 frames. SA-ConvLSTM shows its high efficiency and the flexibility on the KTH dataset. It improves the PSNR of the state-of-the-art model by 0.86 and SSIM by 0.026. Our base model still achieves comparable results with PredRNN.

The prediction samples on KTH are visualized in Figure 5. It is difficult for ConvLSTM to make high-quality predictions. The ConvLSTM with self-attention (base model) achieves a similar prediction performance as PredRNN. Compared to PredRNN, SA-ConvLSTM can provide more texture information, such as black pants and a white coat in the Figure 5. The prediction errors marked by circles and blurry human bodies indicate that ConvLSTM and PredRNN cannot maintain accuracy and image quality when carrying out long-term prediction. In contrast, SA-ConvLSTM can not only keep more texture information but also improve prediction accuracy.

4.5 Attention Visualization

In order to explain the effect of the self-attention mechanism in the proposed SA-ConvLSTM, we randomly choose some examples from the test set of MovingMNIST and visualize the attention maps in Figure 6, where the attention maps are

Table 3: Comparisons to state-of-the-art methods on the TaxiBJ test set (a) and the KTH test set (b) separately.

(a) Per-frame MSE on the TaxiBJ test set. All models predict the next 4 images (traffic conditions for the next two hours) via 4 historical traffic flow images.

Models	Frame 1↓	Frame 2↓	Frame 3↓	Frame 4↓
ST-ResNet (Zhang et al. 2017)	0.460	0.571	0.670	0.762
VPN (Kalchbrenner et al. 2017)	0.427	0.548	0.645	0.721
FRNN (Oliu et al. 2018)	0.331	0.416	0.518	0.619
PredRNN (Wang et al. 2017a)	0.318	0.427	0.516	0.595
PredRNN++ (Wang et al. 2018b)	0.319	0.399	0.500	0.573
MIM (Wang et al. 2019)	0.309	0.390	0.475	0.542
Base Model (Ours)	0.291	0.367	0.460	0.524
SA-ConvLSTM (Ours)	**0.269**	**0.356**	**0.426**	**0.507**

(b) Comparison of the next 20-frame prediction on the KTH test set. PSNR and SSIM are adopted.

Models	PSNR↑	SSIM↑
ConvLSTM (Shi et al. 2015)	23.58	0.712
TrajGRU (Shi et al. 2017b)	26.97	0.790
DFN (Jia et al. 2016)	27.26	0.794
MCNet (Villegas et al. 2017)	25.95	0.804
PredRNN (Wang et al. 2017a)	27.55	0.839
PredRNN++ (Wang et al. 2018b)	28.47	0.865
Base Model (Ours)	27.25	0.837
SA-ConvLSTM (Ours)	**29.33**	**0.891**

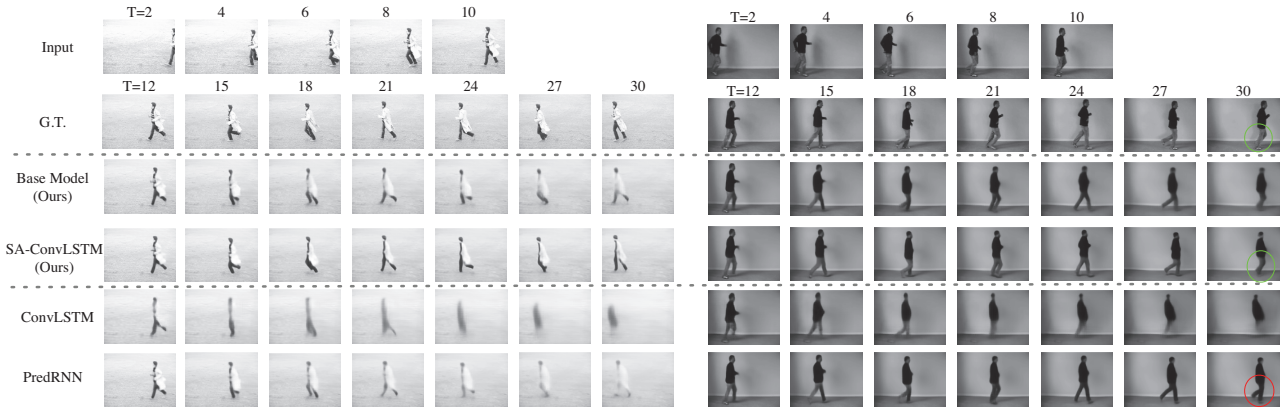

Figure 5: Visualization examples on the KTH test set. Each model predicts the next 20 frames by observing 10 frames. Our SA-ConvLSTM can generate the sharpest and the most precise prediction.

from the first and last layers by querying a specific point "+". The area with warmer color has a more relevant relationship with the query point. When the "+" is on the digits, the attention is concentrated on the foreground, as shown in the "T=13", "T=19" of the second row and "T=10" of the third row. On the contrary, when the query point is on the background, most of the weights focus on the background, as demonstrate in the "T=1" of the second row and "T=16" of the fourth row. The low-level (layer 1) features are shift-invariant, such that the background features are basically the same, and layer 1 can uniformly attends the background pixels. In contrast, the features of layer 4 have more semantic information. Here, the probability of numbers appearing in corners is very low in Moving-MNIST. This kind of statistical prior can be learned by the network. Our SAM learns to transform features at corners to background filters, which can be used to construct more accurate foreground or background features.

5 Conclusion

In this paper, we propose the SA-ConvLSTM for spatiotemporal prediction. Since the prediction of the current time step can benefit from the past relevant features, we construct a self-attention memory module to capture long-range dependencies in terms of spatial and temporal dimensions. We evaluated our models on MovingMNIST and KTH datasets for the multi-frame prediction and TaxiBJ for the traffic flow prediction. Ablation experiments demonstrate the effectiveness of self-attention and the additional memory $\mathcal{M}$ on different types of data. The proposed SA-ConvLSTM achieves the best results on all datasets with much fewer parameters and higher efficiency than the previous state-of-the-art model MIM.

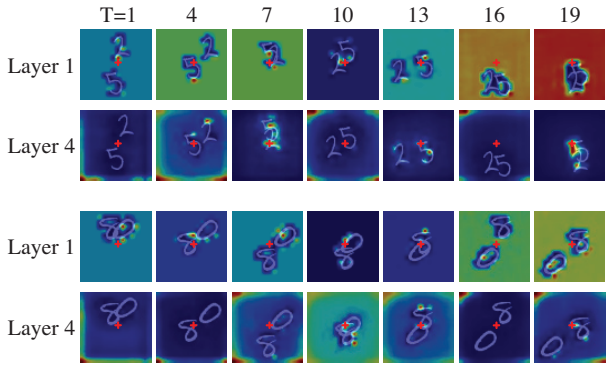

Figure 6: Visualization of attention maps on the MovingMNIST test set. Attention maps in the 1st and the 4th layers are visualized, where "+" is the querying point. Best view in color and warmer color represents the higher correlation.

Acknowledgments

This work is supported by NSFC project Grant No. U1833101, Shenzhen Science and Technologies project under Grant No. JCYJ20160428182137473 and the Joint Research Center of Tencent & Tsinghua.

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bengio, Y.; Simard, P.; Frasconi, P.; et al. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5(2):157–166.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS 2015*, 1171–1179.

Bianchini, M., and Scarselli, F. 2014. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE transactions on neural networks and learning systems* 25(8):1553–1565.

Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; and Feng, J. 2018. Aˆ 2-nets: Double attention networks. In *NIPS 2018*, 352–361.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *CVPR 2017*, 1251–1258.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *ICCV 2017*.

Finn, C.; Goodfellow, I.; and Levine, S. 2016. Unsupervised learning for physical interaction through video prediction. In *NIPS 2016*, 64–72.

Fu, J.; Liu, J.; Tian, H.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *CVPR 2019*.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV 2019*.

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Gool, L. V. 2016. Dynamic filter networks. In *NIPS 2016*, 667–675.

Kalchbrenner, N.; Oord, A.; Simonyan, K.; Danihelka, I.; Vinyals, O.; Graves, A.; and Kavukcuoglu, K. 2017. Video pixel networks. In *ICML 2017*, 1771–1779.

Lerer, A.; Gross, S.; and Fergus, R. 2016. Learning physical intuition of block towers by example. In *ICML 2016*, 430–438.

Luo, W.; Li, Y.; Urtasun, R.; and Zemel, R. 2016. Understanding the effective receptive field in deep convolutional neural networks. In *NIPS 2016*, 4898–4906.

Oliu, M.; Selva, J.; Escalera, S.; and Escalera, S. 2018. Folded recurrent neural networks for future video prediction. In *ECCV 2018*, 716–731.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. In *ICML 2013*, 1310–1318.

Schuldt, C.; Laptev, I.; Caputo, B.; and Caputo, B. 2004. Recognizing human actions: a local svm approach. In *ICPR 2004*, volume 3, 32–36.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS 2015*, 802–810.

Shi, X.; Gao, Z.; Lausen, L.; Wang, H.; Yeung, D.-Y.; Wong, W.-k.; and Woo, W.-c. 2017a. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NIPS 2017*, 5622–5632.

Shi, X.; Gao, Z.; Lausen, L.; Wang, H.; Yeung, D.-Y.; Wong, W.-k.; and Woo, W.-c. 2017b. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NIPS 2017*, 5617–5627.

Srivastava, N.; Mansimov, E.; Salakhudinov, R.; Salakhudinov, R.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *ICML 2015*, 843–852.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS 2017*, 5998–6008.

Villegas, R.; Yang, J.; Hong, S.; Lin, X.; and Lee, H. 2017. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13(4):600–612.

Wang, Y.; Long, M.; Wang, J.; Gao, Z.; and Philip, S. Y. 2017a. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NIPS 2017*, 879–888.

Wang, Y.; Long, M.; Wang, J.; Gao, Z.; and Philip, S. Y. 2017b. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NIPS 2017*, 879–888.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018a. Non-local neural networks. In *CVPR 2018*, 7794–7803.

Wang, Y.; Gao, Z.; Long, M.; Wang, J.; and Philip, S. Y. 2018b. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *ICML 2018*, 5110–5119.

Wang, Y.; Zhang, J.; Zhu, H.; Long, M.; Wang, J.; and Yu, P. S. 2019. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *CVPR 2019*, 9154–9162.

Xu, Z.; Wang, Y.; Long, M.; and Wang, J. 2018. Predcnn: predictive learning with cascade convolutions. In *IJCAI 2018*, 2940–2947.

Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; and Yi, X. 2016. Dnn-based prediction model for spatio-temporal data. In *ACM SIGSPATIAL 2016*, 92.

Zhang, J.; Zheng, Y.; Qi, D.; and Qi, D. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI 2017*.

Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2019. Self-attention generative adversarial networks. In *ICML 2019*, 7354–7363.