LONG HISTORY SHORT-TERM MEMORY
FOR LONG-TERM VIDEO PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

While video prediction approaches have advanced considerably in recent years, learning to predict long-term future is challenging — ambiguous future or error propagation over time yield blurry predictions. To address this challenge, existing algorithms rely on extra supervision (e.g., action or object pose), motion flow learning, or adversarial training. In this paper, we propose a new recurrent unit, Long History Short-Term Memory (LH-STM). LH-STM incorporates long history states into a recurrent unit to learn longer range dependencies. To capture spatio-temporal dynamics in videos, we combined LH-STM with the Context-aware Video Prediction model (ContextVP). Our experiments on the KTH human actions and BAIR robot pushing datasets demonstrate that our approach produces not only sharper near-future predictions, but also farther into the future compared to the state-of-the-art methods.

1 INTRODUCTION

Learning the dynamics of an environment and predicting consequences in the future has become an important research problem. A common task is to train a model that accurately predicts pixel-level future frames conditioned on past frames. It can be utilized for intelligent agents to guide them to interact with the world, or for other video analysis tasks such as activity recognition. An important component of designing such models is how to effectively learn good spatio-temporal representations from video frames. The Convolutional Long Short-Term Memory (ConvLSTM) network (Xingjian et al., 2015) has been a popular model architecture choice for video prediction. However, recent state-of-the-art approaches produce high-quality predictions only for one or less then ten frames (Lotter et al., 2016; Villegas et al., 2017a; Byeon et al., 2018). Learning to predict long-term future video frames remains challenging due to 1) the presence of complex dynamics in high-dimensional video data, 2) prediction error propagation over time, and 3) inherent uncertainty of the future.

In order to predict reasonably long-term video frames, some approaches represent video dynamics by hierarchical methods (Wichers et al., 2018; Villegas et al., 2017b) or action-conditioned models (Finn et al., 2016; Oh et al., 2015). To efficiently model video dynamics, spatio-temporal ConvLSTM models have also been studied (Wang et al., 2017; 2018a;b). Another popular direction is to learn multiple plausible futures via an adversarial loss (Mathieu et al., 2015) or a Variational Autoencoder (VAE) (Denton & Fergus, 2018; Babaeizadeh et al., 2017; Lee et al., 2018). In this paper, we attempt to solve two of the issues mentioned above: learning complex dynamics of videos and minimizing long-term prediction error by fully observing the history.

We propose a novel modification of the ConvLSTM structure, Long History Short-Term Memory (LH-STM). LH-STM learns to interconnect history states and the current input by a History Soft-Selection Unit (HistSSel) and double memory modules. The weighted history states computed by our HistSSel units are combined with the history memory and then used to update the current states by the update memory. The proposed method brings the power of higher-order RNNs (Soltani & Jiang, 2016) to ConvLSTMs, which have been limited to simple recurrent mechanisms so far. The HistSSel unit acts as a short-cut to the history, so the gradient flow in the LSTM is improved. More powerful RNNs are likely to be necessary to solve the hard and unsolved problem of long-term video prediction, which is extremely challenging for current architectures. Moreover, by disentangling the history and update memories, our model can fully utilize the long history states.
This structure can better model long-term dependencies in sequential data. In this paper, the proposed modification is integrated into the ConvLSTM-based architectures to solve the long-term video prediction problem: Context-aware Video Prediction (ContextVP) model. The proposed models can fully leverage long-range spatio-temporal contexts in real-world videos. Our experiments on the KTH human actions and the BAIR robot pushing datasets show that our model produces sharp and realistic predictions for more frames into the future compared to recent state-of-the-art long-term video prediction methods.

2 Related Work

Learning Long-Term Dependencies with Recurrent Neural Networks (RNN): While Long Short-Term Memory (LSTM) has been successful for sequence prediction, many recent approaches aim to capture longer-term dependencies in sequential data. Several works have proposed to allow dynamic recurrent state updates or to learn more complex transition functions. Chung et al. (2016) introduced the hierarchical multiscale RNN that captures a hierarchical representation of a sequence by encoding multiple time scales of temporal dependencies. Koutnik et al. (2014) modified the standard RNN to a Clockwork RNN that partitions hidden units and processes them at different clock speeds. Neil et al. (2016) introduced a new time gate that controls update intervals based on periodic patterns. Campos et al. (2017) proposed an explicit skipping module for state updates. Zilly et al. (2017a) increased the recurrent transition depth with highway layers. Fast-Slow Recurrent Neural Networks (Mujika et al., 2017) incorporate ideas from both multiscale (Schmidhuber, 1992; El Hihi & Bengio, 1996; Chung et al., 2016) and deep transition (Pascanu et al., 2013; Zilly et al., 2017) RNNs. The advantages of the above approaches are efficient information propagation through time, better long memory traces, and generalization to unseen data.

Alternative solutions include the use of history states, an attention model or skip connections. Soltani & Jiang (2016) investigated a higher order RNN to aggregate more history information and showed that it is beneficial for long range sequence modeling. Cheng et al. (2016) deployed an attention mechanism in LSTM to induce relations between input and history states. Gui et al. (2018a) incorporated into an LSTM, dynamic skip connections and reinforcement learning to model long-term dependencies. These approaches use the history states in a single LSTM by directly adding more recurrent connections or adding an attention module in the memory cell. These models are used for one-dimensional sequence modeling, whereas our proposed approach separates the history and update memories that learn to encode the long-range relevant history states. Furthermore, our approach is more suitable for high-dimensional (e.g., video) prediction tasks.

Video Prediction: The main issue in long-term pixel-level video prediction is how to capture long-term dynamics and handle uncertainty of the future while maintaining sharpness and realism. Oh et al. (2015) introduced action-conditioned video prediction using a Convolutional Neural Network (CNN) architecture. Villegas et al. (2017b) and Wichers et al. (2018) focused on a hierarchical model to predict long-term videos. Their model estimates high-level structure before generating pixel-level predictions. However, the approach by Villegas et al. (2017b) requires object pose information as ground truth during training. Finn et al. (2016) used ConvLSTM to explicitly model pixel motions. To generate high-quality predictions, many approaches train with an adversarial loss (Mathieu et al., 2015; Wichers et al. 2018; Vondrick et al., 2016; Vondrick & Torralba, 2017; Denton et al., 2017; Lee et al., 2018). Another active line of investigation is to train stochastic prediction models using VAEs (Denton & Fergus, 2018; Babaeizadeh et al., 2017; Lee et al., 2018). These models predict plausible futures by sampling latent variables and produce long-range future predictions.

The Spatio-temporal LSTM (Wang et al., 2017; 2018a) was introduced to better represent the dynamics of videos. This model is able to learn spatial and temporal representations simultaneously. Byeon et al. (2018) introduced a Multi-Dimensional LSTM-based approach (Stollenga et al., 2015) for video prediction. It contains directional ConvLSTM-like units that efficiently aggregate the entire spatio-temporal contextual information. Wang et al. (2018b) recently proposed a memory recall function with 3D ConvLSTM. This work is the most related to our approach. It uses a set of cell states with an attention mechanism to capture the long-term frame interaction similar to the work of Cheng et al. (2016). With 3D-convolutional operations, the model is able to capture short-term and long-term information flow. In contrast to this work, the attention mechanism in our model is used for a set of hidden state. We also disentangle the history and update memory cells to better
memorize and restore the relevant information. By integrating a double memory LH-STM into the context-aware video prediction (ContextVP) model \cite{Byeon2018}, our networks can capture the entire spatio-temporal context for a long-range video sequence.

3 Method

In this section, we first describe the standard ConvLSTM architecture and then introduce the LH-STM. Finally, we explain the ConvLSTM-based network architectures for multi-frame video prediction using the Context-aware Video Prediction model (ContextVP) \cite{Byeon2018}.

3.1 Convolutional LSTM

Let $X^n = \{X_1, ..., X_n\}$ be an input sequence of length $n$. $X_k \in \mathbb{R}^{h \times w \times c}$ is the $k$-th frame, where $k \in \{1, ..., n\}$, $h$ is the height, $w$ the width, and $c$ the number of channels. For the input frame $X_k$, a ConvLSTM unit computes the current cell and hidden states $(C_k, H_k)$ given the cell and hidden states from the previous frame, $(C_{k-1}, H_{k-1})$:

$$C_k, H_k = \text{ConvLSTM}(X_k, H_{k-1}, C_{k-1}),$$

by computing the input, forget, output gates $i_k, f_k, o_k$, and the transformed cell state $\hat{C}_k$:

$$i_k = \sigma(W_i \ast X_k + M_i \ast H_{k-1} + b_i),$$
$$f_k = \sigma(W_f \ast X_k + M_f \ast H_{k-1} + b_f),$$
$$o_k = \sigma(W_o \ast X_k + M_o \ast H_{k-1} + b_o),$$
$$\hat{C}_k = \tanh(W_c \ast X_k + M_c \ast H_{k-1} + b_c),$$
$$C_k = f_k \odot C_{k-1} + i_k \odot \hat{C}_k,$$
$$H_k = o_k \odot \tanh(C_k),$$

where $\sigma$ is the sigmoid function, $W$ and $M$ are 2D convolutional kernels for input-to-state and state-to-state transitions, $(\ast)$ is the convolution operation, and $(\odot)$ is element-wise multiplication. The size of the weight matrices depends on the size of convolutional kernel and the number of hidden units.

3.2 Long History Short-Term Memory (LH-STM)

LH-STM is an extension of the standard ConvLSTM model by integrating a set of history states into the LSTM unit. The history states include the spatio-temporal context of each frame in addition to the pixel level information in the frame itself. Figure 1 illustrates the differences between a standard RNN, Higher-Order RNN \cite{Soltani2016}, and our proposed model (Double and Single LH-STM).

![Graphical illustrations of the standard RNN, Higher-Order RNN, Single and Double LH-STM](image)

**Figure 1:** Graphical illustrations of the standard RNN, Higher-Order RNN \cite{Soltani2016}, Single and Double LH-STM. $X_k$ and $\hat{X}_k$, respectively indicate the input and predicted frame at time $k$. $H_{k-3}, H_{k-2}, H_{k-1}$ are past hidden states. A blue line indicates a recurrent connection; green color indicates History Soft-Selection unit, and a circle donate a memory unit.

**History Soft-Selection Unit (HistSSel):** The HistSSel unit computes the relationship between the recent history and the earlier ones using dot-product similar to \cite{Vaswani2017}.

\footnote{The purpose of this unit is to compute the importance of each history state. While we used a self-attention-like unit in this paper, this can be achieved with other common layers as well, e.g., fully connected or convolutional layers.}
mechanism can be formulated as $\text{SoftSel}(Q, K, V) = \text{softmax}(W^Q Q \cdot W^K K) \cdot W^V V$. It consists of queries (Q), keys (K) and values (V). It computes the dot products of the queries and the keys; and then applies the a softmax function. Finally, the values (V) are weighted by the outputs of softmax function. The queries, keys, and values can be optionally transformed by the $W^Q$, $W^K$, and $W^V$ matrices.

Using this mechanism, HistSSel computes the relationship between the last hidden state $H_{k-1}$ and the earlier hidden states $H_{k-m:k-2}$ at time step $k$ (See Figure 2). $H_{k-m:k-2}$ is the set of previous hidden states, $(H_{k-m}, H_{k-m-1}, \cdots H_{k-3}, H_{k-2})$. We will show in Section 4.2 the benefit of using history states versus using the past input frames directly. The history soft-selection mechanism can be formulated as follows:

$$\text{HistSSel}(H_{k-1}, H_{k-m:k-2}) = \sum_{i=2}^{n} \text{softmax}_i(\tilde{H}^Q_{k-1} \cdot \tilde{H}^K_{k-i}) \cdot \tilde{H}^V_{k-i},$$  

$$\tilde{H}^j_{i} = W^j_i H_i + b^j_i, j \in \{Q, K, V\}$$ \hspace{1cm} (3)

**Single LH-STM:** A simple way to employ HistSSel in ConvLSTM (Equation 1) is to add HistSSel($H_{k-1}, H_{k-m:k-2}$) in addition to the input and the previous states ($X_k, H_{k-1}, C_{k-1}$) in the Equation 2 Figure 1c shows a diagram of the computation. This direct extension is named Single LH-STM: $H_k = \text{SingleConvLSTM}(X_k, H_{k-1}, C_{k-1}, \text{HistSSel}(H_{k-1}, H_{k-m:k-2}))$.

$$i_k = \sigma(W_i * X_k + M_i * H_{k-1} + \text{HistSSel}(H_{k-1}, H_{k-m:k-2}) + b_i),$$

$$f_k = \sigma(W_f * X_k + M_f * H_{k-1} + \text{HistSSel}(H_{k-1}, H_{k-m:k-2}) + b_f),$$

$$o_k = \sigma(W_o * X_k + M_o * H_{k-1} + \text{HistSSel}(H_{k-1}, H_{k-m:k-2}) + b_o)$$

$$\hat{C}_k = \tanh(W_c * X_k + M_c * H_{k-1} + \text{HistSSel}(H_{k-1}, H_{k-m:k-2}) + b_c),$$

$$C_k = i_k \odot \hat{C}_k + f_k \odot C_{k-1},$$

$$H_k = o_k \odot \tanh(C_k).$$ \hspace{1cm} (4)

**Double LH-STM:** To effectively learn dynamics from the history states, we propose Double LH-STM. It contains two ConvLSTM blocks, History LSTM (H-LSTM) and Update LSTM (U-LSTM). The goal of the Double LH-STM is to explicitly separate the long-term history memory and the update memory. By disentangling these, the model can better encode complex long-range history and keep track of the their dependencies. Figure 1d illustrates a diagram of Double LH-STM.
The H-LSTM block explicitly learns the complex transition function from the (possibly entire) set of past hidden states, \( H_{k,m:k-1} \), where \( 1 < m < k \). If \( m = k \), H-LSTM incorporates the entire history up to the time step \( k - 1 \). The History LSTM can be formulated as:

\[
\begin{align*}
    i'_{k-1} &= \sigma(M'_i \ast H_{k-1} + \text{HistSSel}(H_{k-1}, H_{k-m:k-2}) + b'_i), \\
    f'_{k-1} &= \sigma(M'_f \ast H_{k-1} + \text{HistSSel}(H_{k-1}, H_{k-m:k-2}) + b'_f), \\
    o'_{k-1} &= \sigma(M'_o \ast H_{k-1} + \text{HistSSel}(H_{k-1}, H_{k-m:k-2}) + b'_o), \\
    \hat{C}'_{k-1} &= \tanh(M'_c \ast H_{k-1} + \text{HistSSel}(H_{k-1}, H_{k-m:k-2}) + b'_c), \\
    C'_{k-1} &= f'_{k-1} \odot C_{k-2} + i'_{k-1} \odot \hat{C}'_{k-1}, \\
    H'_{k-1} &= o'_{k-1} \odot \tanh(c_{k-1})'.
\end{align*}
\]  

(5)

The U-LSTM block updates the states \( H_k \) and \( C_k \) for the time step \( k \), given the input \( X_k \), previous cell state \( C_{k-1} \), and the output of the H-LSTM \( H'_{k-1} \), by replacing \( H_{k-1} \) with \( H'_{k-1} \) in Equation 2.

### 3.3 Implementation Details

**Model Architectures:** ConvLSTM (Xingjian et al. [2015] Stollenga et al. [2015]) is a popular building block for spatio-temporal sequence forecasting problems. In this paper, we focused on the recent state-of-the-art ConvLSTM-based architectures proposed by Byeon et al. (2018): the context-aware video prediction model (ContextVP-4). It can capture sufficient spatio-temporal context for video prediction and achieve state-of-the-art results on KTH human actions and BAIR robot pushing dataset. In this paper, the ConvLSTM block is replaced with LH-STM. Additionally, we exploit two types of skip connections to avoid the problem of vanishing gradients (Hochreiter et al. [2001]) and allow long-term prediction: (a) between previous and the current recurrent layers (blue dotted line in Figure 3) and (b) across layers (green solid line in Figure 3). See Appendix B for details of the network architectures.

ContextVP-4 (Byeon et al. [2018]) consists of 4 layers of five directional Parallel Multi-Dimensional LSTM (PMD) units (Stollenga et al. [2015]) along the \( b+, h-, w+, w- \), and \( t\) recurrence directions. A PMD unit along the time direction \((-t)\) is mathematically equivalent to a standard ConvLSTM (Equation 2). By using LSTM connectivities across the temporal and spatial dimensions, each processing layer covers the entire context in a video. Following Byeon et al. (2018), we also included weighted blending, directional weight sharing (DWS), and two skip connections between the layers 1-3 and 2-4 (see Figure 3). See Appendix B for details of the network architectures.

**Skip connection in a recurrent block (R-Skip):** A skip connection is commonly used in LSTM layers (Melis et al. [2017] Merity et al. [2017] Wang et al. [2018a]) for long-term gradient propagation. Our proposed model concatenates outputs of the previous and current LSTM layers (see Figure 3).

**Gated skip connection across layers (L-Skip):** Unlike in Byeon et al. (2018). We added the gated skip connection across layers. In our model, the multiplicative gate is added to control the flow of information across layers. It stabilizes learning as depth increases. A layer is formulated as:

\[
Y = H_l \cdot T_l + X_{l-1} \cdot (1 - T_l), \quad i < l,
\]

where \( H_l = X_l \cdot W^H_l + b^H \) and \( T_l = \sigma(X_l \cdot W^T_l + b^T) \). \( X_l \) is the input at the \( l \)th layer, \( T \) is a gate, and \( H_l \) is the transformation of the input at the layer \( l \).

**Loss:** The loss function to train the networks is the combination of \( L_1 \)-norm, \( L_2 \)-norm, and perceptual loss (Johnson et al. [2016] Zhang et al. [2018]):

\[
\mathcal{L}(Y, \hat{X}) = \lambda_1 L_1(Y, \hat{X}) + \lambda_2 L_2(Y, \hat{X}) + \lambda_{pl} L_{pl}(Y, \hat{X}),
\]

(7)

where \( Y \) and \( \hat{X} \) are the target and the predicted frames, respectively. \( \lambda \) is a weight for each function. \( L_p \)-norm is defined as \( \mathcal{L}_p(Y, \hat{X}) = \| Y - \hat{X} \|_p \), \( p = 1 \) for \( L_1 \)-norm, and \( p = 2 \) for \( L_2 \)-norm. The perceptual loss computes the cosine distance between the feature maps extracted from a VGG-16 network pre-trained on the ImageNet dataset (Simonyan & Zisserman [2014]) as \( L_{pl}(y, \hat{y}) = 1 - \frac{1}{d} \sum_{r \in \text{w}^l} \sum_{m \in \text{w}^l} (\phi(y)_r \cdot \phi(\hat{y})_r) \), where \( \phi(y)_r \) and \( \phi(\hat{y})_r \) are the feature maps of the target and the predicted frames \( Y \) and \( \hat{X} \) respectively at the layer \( l \). The size of the feature map at layer \( l \) is \( h^l \times w^l \).
Table 1: Evaluation of multi-frame prediction on the KTH human actions dataset (resolution $128 \times 128$). All models are trained for 10 frame prediction given 10 input frames. They are then tested to recursively predict 40 future frames. The reported metrics are averages across the predicted frames. Higher values indicate better results. (*) We re-trained the model using the provided code by Wang et al. (2018a) as their originally reported numbers are not with the same experimental setup as ours. See Appendix D for training details.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>VIF</th>
<th># parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvLSTM (Xingjian et al., 2015)</td>
<td>22.85</td>
<td>0.639</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FRNN (Oliu et al., 2018)</td>
<td>23.77</td>
<td>0.678</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PredRNN (Wang et al., 2017)</td>
<td>24.16</td>
<td>0.703</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PredRNN++ (original) (Wang et al., 2018a)</td>
<td>25.21</td>
<td>0.741</td>
<td>-</td>
<td>15.1M</td>
</tr>
<tr>
<td>PredRNN++ (retrained) (Wang et al., 2018a)</td>
<td>26.37</td>
<td>0.734</td>
<td>-</td>
<td>15.1M</td>
</tr>
<tr>
<td>VAE (retrained) (Lee et al., 2018)</td>
<td>26.23</td>
<td>0.777</td>
<td>0.289</td>
<td>9.8M</td>
</tr>
<tr>
<td>Deterministic (retrained) (Lee et al., 2018)</td>
<td>27.59</td>
<td>0.796</td>
<td>0.328</td>
<td>8.9M</td>
</tr>
<tr>
<td>E3D-LSTM (Wang et al., 2018b)</td>
<td>27.24</td>
<td>0.810</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ContextVP-4 (base model) (Byeon et al., 2018)</td>
<td>26.37</td>
<td>0.785</td>
<td>0.320</td>
<td>17.3M</td>
</tr>
<tr>
<td>Single LH-STM (ours)</td>
<td>27.83</td>
<td>0.816</td>
<td>0.354</td>
<td>18.6M</td>
</tr>
<tr>
<td>Double LH-STM (ours)</td>
<td><strong>28.64</strong></td>
<td><strong>0.826</strong></td>
<td><strong>0.367</strong></td>
<td><strong>16.2M</strong></td>
</tr>
</tbody>
</table>

4 EXPERIMENTS

We evaluated our LH-STM model on the KTH human actions (Laptev et al., 2004) and BAIR action-free robot pushing (Ebert et al., 2017) datasets. Videos in the KTH human actions dataset consist of six human actions including walking, jogging, running, boxing, hand waving, and hand clapping. The BAIR action-free robot pushing dataset includes randomly moving robotic arms, which push objects on a table. The videos have a resolution of $64 \times 64$ pixels. We show quantitative comparisons by computing frame-wise Structural Similarity (SSIM) (Wang et al., 2004), Peak Signal-to-Noise Ratio (PSNR), and Visual Information Fidelity (VIF) (Sheikh & Bovik, 2005) between the ground truth and generated video sequences.

4.1 KTH HUMAN ACTIONS DATASET

We follow the experimental setup of Lee et al. (2018). We evaluate our methods on $128 \times 128$ pixels (Table 1). We train all models to predict the next 10 frames, given 10 prior ones. For testing, all models recursively predict 40 frames.

Our Double LH-STM outperforms all competing methods on all metrics including the base model (ContextVP). We also compared per-frame PSNR and SSIM scores in Figure 5. Our Double LH-STM model consistently outperforms other methods for the 40 predicted frames. The improvement is larger for longer-term frame prediction into the future (40 frames). Qualitative results can be found in Figure 8 and Appendix C (Figure 10).

Our models are compared with FRNN (Oliu et al., 2018), PredRNN (Wang et al., 2017), PredRNN++ (Wang et al., 2018a), E3D-LSTM (Wang et al., 2018b), VAE and deterministic models (Lee et al., 2018). Our model outperforms all other methods.

4.2 STATE HISTORY VS INPUT HISTORY

The proposed LH-STM models use past hidden states. Alternatively, input frames can be directly applied to LH-STM as illustrated in Figure 4. We compared the performance of using input and state history with the same LH-STM models and the same model size in Table 2 and Figure 5.

---

*We note that Lee et al. (2018) reported the performance of VAE+GAN, VAE only, GAN only, and of their deterministic models. We select their best (VAE only) and the most comparable (deterministic) models to compare against. To evaluate the VAE model, we take 100 random input samples for each video following Lee et al. (2018). The reported numbers in this paper are for their method’s “median” output, among the 100 randomly generated ones, compared to the ground truth.*
(Double LH-STM w/ input history) on the KTH human actions dataset. Figure 8 shows a qualitative comparison of using state history (Double LH-STM) and input history (Double, input history). The results show the importance of using history states with LH-STM and that the state history is more suitable for videos.

Figure 4: The use of input history with LH-STM Single (left) and Double (right)

![Diagram](image)

Table 2: Comparison of using input vs state history with LH-STM Single and Double on the KTH human actions dataset (resolution $128 \times 128$).

<table>
<thead>
<tr>
<th>History</th>
<th>LH-STM</th>
<th>PSNR</th>
<th>SSIM</th>
<th>VIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Single</td>
<td>25.78</td>
<td>0.779</td>
<td>0.302</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>26.45</td>
<td>0.788</td>
<td>0.308</td>
</tr>
<tr>
<td>State</td>
<td>Single</td>
<td>27.83</td>
<td>0.816</td>
<td>0.354</td>
</tr>
<tr>
<td></td>
<td>Double</td>
<td>28.64</td>
<td>0.826</td>
<td>0.367</td>
</tr>
</tbody>
</table>

4.3 Longer Prediction

To test the performance of longer-term prediction (more than 40 frames), we additionally trained the KTH human actions dataset with a resolution of $64 \times 64$ pixels. Figure 6 compares the performance of 80 frame prediction with Double LH-STM, Single, the base model (ContextVP), as well as two state-of-the-art models (VAE and Deterministic (Lee et al., 2018)). The model size and the experimental setup are the same as the experiments of a resolution $128 \times 128$ (Section 4.1). From this comparison, we can observe that the performance of Double LH-STM does not degrade over a long period of time. Notably, the performance of our model at 80 frames is similar to that of the baseline at $\approx 30$ frames.

![Graph](image)
Figure 8: Qualitative comparison of Double LH-STM, Single LH-STM, Double LH-STM with input history (See Section 4.2 for the details), the base model (ContextVP (Byeon et al., 2018)), and the state-of-the-art model (Deterministic (Lee et al., 2018)). Double LH-STM in overall produces the sharper images and greater realism compared to other models. Compare to the Deterministic model (Lee et al., 2018), our model can capture more precise motion.

Table 3: Evaluation of multi-frame prediction on the BAIR robot pushing dataset. All models are trained for 10 frame prediction given 5 input frames. They are then tested to predict 25 future frames. The reported numbers are averages across the predicted frames. Higher values indicate better results.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>VIF</th>
<th># parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE (Lee et al., 2018)</td>
<td>17.32</td>
<td>0.779</td>
<td>0.524</td>
<td>6.7M</td>
</tr>
<tr>
<td>Deterministic (Lee et al., 2018)</td>
<td>18.08</td>
<td>0.801</td>
<td>0.544</td>
<td>6.4M</td>
</tr>
<tr>
<td>ContextVP-4 (Byeon et al., 2018)</td>
<td>18.38</td>
<td>0.818</td>
<td>0.519</td>
<td>17.3M</td>
</tr>
<tr>
<td>Single LH-STM (ours)</td>
<td>18.62</td>
<td>0.814</td>
<td>0.514</td>
<td>17.6M</td>
</tr>
<tr>
<td>Double LH-STM (ours)</td>
<td><strong>18.81</strong></td>
<td><strong>0.824</strong></td>
<td>0.541</td>
<td>16.3M</td>
</tr>
</tbody>
</table>

4.4 BAIR ROBOT PUSHING DATASET

Our experimental setup follows that of Lee et al. (2018). All models are conditioned on five frames to predict the next ten frames. The models are tested to predict 25 frames. We evaluated our LH-STM models (Single and Double) and compared them with the base model (ContextVP), the VAE, and deterministic models (Lee et al., 2018) in Table 3. Our Double LH-STM achieved better PSNR and SSIM scores than the deterministic and VAE models. It can be seen from the per-frame prediction results (Figure 7), that the performance LH-STM is higher over time. We found that this dataset contains diverse random motion types which makes the long-term prediction task challenging. Qualitative results can be found in Appendix C (Figure 11).

5 CONCLUSION

In this paper, we have introduced a new RNN structure, LH-STM, which incorporates long history states into the recurrent unit to learn longer range dependencies. This module is combined with the ContextVP architecture allowing context aware long-term video prediction. The resulting model produces more realistic predictions for a longer period of time compared to the recent state-of-the-art methods. The proposed approach can easily be extended to standard LSTM or Gated Recurrent Units (GRU) and applied to tasks that require a long-term memory trace, e.g., language modeling or speech recognition.

Lee et al. (2018) used two input frames. We increase the number of input frames as motion in two frames is almost not visible. We increased the number of input frames to provide enough history to the model. We also re-trained the models of Lee et al. (2018) with five input frames.
REFERENCES


Under review as a conference paper at ICLR 2020

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. 2018b.


A NETWORK DETAILS

Each layer of ContextVP-4 contains five Parallel Multi-Dimensional LSTM (PMD) units (see Figure 9). The parameters of opposite directions are shared. We followed the same network architecture as Byeon et al. (2018) except for the number of neurons.

\[
S_k = [H_k^{h+}, H_k^{h-}, H_k^{w+}, H_k^{w-}, H_k^{t-}],
\]
\[
M_k = f(W_k \cdot S_k + b_k),
\]

where \( H \) is the output of the PMD units from \( h+, h-, w+, w- \), and \( t- \) directions, and \( W \) is the weight matrix.

**Weighted Blending:** This layer learns the relative importance of each direction during training.

**Directional Weight-Sharing:** Weights and biases of the PMD units in opposite directions (i.e., the \( h+ \) and \( h- \) directions, \( w+ \) and \( w- \) directions) are shared.

We adjusted the number of neurons, so all models have similar model size (\( \approx 17M \)). LH-STM Double contains 64-64-128-128 neurons for the input size of 64 \( \times \) 64 pixels and 64-128-128-128 for 128 \( \times \) 128. LH-STM Single contains 128-128-128-128 neurons for the input size of 64 \( \times \) 64 pixels and 64-128-128-256 for 128 \( \times \) 128.

The initial learning rate is \( 1e-4 \) found by grid search between \( 1e-3 \) and \( 1e-5 \). The learning rate is decayed every 10 epochs with the decay rate 0.95. Weights are initialized using the Xavier’s normalized initializer (Glorot & Bengio, 2010) and the recurrent states are all initialized to zero.

B ABLATION STUDY

We performed a series of ablation studies to evaluate the importance of each component of our proposed model 1) RNN type (standard ConvLSTM, Single LH-STM, or Double LH-STM); 2) different losses (\( L_1 \), \( L_2 \), and the perceptual loss); adding skip connections 3) in recurrent blocks; and 4) across layers (see Section 3.3 for details). We evaluated performance on the KTH human actions dataset with a resolution of 64 \( \times \) 64 pixels. The weights for the loss, \( \lambda_1 \), \( \lambda_2 \), and \( \lambda_{pl} \), are tested between zero to one. The results of this experiment are summarized in Table 4. We found that performance improves by adding skip connections in recurrent blocks and across layers. Double LH-STM outperforms the standard ConvLSTM and Single LH-STM. Combination of \( L_1 \), \( L_2 \), and \( L_{pl} \) also improves the performance (with \( \lambda_1 = 1 \), \( \lambda_2 = 1 \), \( \lambda_{pl} = 1 \)).

C QUALITATIVE RESULTS

Figure 10 shows prediction results on the KTH human actions dataset, and Figure 11 presents the results on the BAIR action free robot pushing dataset.
Table 4: An ablation study on KTH human actions dataset. The frame resolution is 64 × 64 pixels. All models are trained to predict 10 future frames given 10 input frames. They are then tested to recursively predict 40 future frames. The reported numbers are averages over all the predicted frames.

<table>
<thead>
<tr>
<th>RNN type</th>
<th>Loss</th>
<th>R-Skip</th>
<th>L-Skip</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvLSTM</td>
<td>(L_1)</td>
<td>✓</td>
<td>✓</td>
<td>27.74</td>
<td>0.812</td>
</tr>
<tr>
<td></td>
<td>(L_1)</td>
<td>✓</td>
<td>✓</td>
<td>27.96</td>
<td>0.812</td>
</tr>
<tr>
<td></td>
<td>(L_1)</td>
<td>✓</td>
<td>✓</td>
<td>28.41</td>
<td>0.812</td>
</tr>
<tr>
<td>Single LH-STM</td>
<td>(L_1)</td>
<td>✓</td>
<td>✓</td>
<td>28.59</td>
<td>0.818</td>
</tr>
<tr>
<td></td>
<td>(L_1) + (L_{pl})</td>
<td>✓</td>
<td>✓</td>
<td>28.46</td>
<td>0.817</td>
</tr>
<tr>
<td>Double LH-STM</td>
<td>(L_1)</td>
<td>✓</td>
<td>✓</td>
<td>28.69</td>
<td>0.812</td>
</tr>
<tr>
<td></td>
<td>(L_1) + (L_{pl})</td>
<td>✓</td>
<td>✓</td>
<td>28.20</td>
<td>0.811</td>
</tr>
<tr>
<td></td>
<td>(L_1) + (L_2)</td>
<td>✓</td>
<td>✓</td>
<td>29.01</td>
<td>0.818</td>
</tr>
<tr>
<td></td>
<td>(L_1) + (L_2) + (L_{pl})</td>
<td>✓</td>
<td>✓</td>
<td>\textbf{29.05}</td>
<td>\textbf{0.820}</td>
</tr>
</tbody>
</table>

D Retraining PredRNN++

As mentioned in the paper, we re-trained PredRNN++ on the KTH human actions dataset using the provided code by [Wang et al.] (2018a) (https://github.com/Yunbo426/predrnn-pp). We conducted grid hyper-parameter search for the learning rate and the patch size. We found in the code that there is an option to divide the input frames into smaller patches. The learning rate search was performed in the range between \(1e^{-3}\) and \(1e^{-4}\). The range of patch size was 1, 2, 4 for both frame resolutions (64 × 64 and 128 × 128 pixels). The reported results in the paper are for a learning rate of \(1e^{-4}\) and the patch size 1 (full resolution).
Figure 10: Qualitative comparison of Double LH-STM, Single LH-STM, Double LH-STM with input history (See Section 4.2 for the details), the base model (ContextVP (Byeon et al., 2018)), and the state-of-the-art model (Deterministic Lee et al. (2018)) on the KTH human actions dataset. Top row shows the ground truth, and the rest is the prediction results. The model is trained for 10 frames and predicted for 40 frames given 10 frames. Double LH-STM produces sharper prediction of human and provide better motion prediction compare to all other models.
Figure 11: Qualitative comparison of Double LH-STM, Single LH-STM, and the base model (ContextVP [Byeon et al., 2018]) on the BAIR robot pushing dataset. Top row shows the ground truth, and the rest is the prediction results. The model is trained for 10 frames and predicted for 25 frames given 5 frames. Double LH-STM can better preserve the shape of the robot arm and produces sharper background prediction.