International Journal of Geo-Information

MDPI

Article

CostNet: A Concise Overpass Spatiotemporal Network for Predictive Learning

Fengzhen Sun [1], Shaojie Li [1,2], Shaohua Wang [3,*], Qingjun Liu [4] and Lixin Zhou [2]

1 Future GIS Laboratory, SuperMap Software Co., Ltd., Beijing 100015, China; sunfengzhen@supermap.com (F.S.); lishaojie@pku.edu.cn (S.L.)
2 School of Software and Microelectronics, Peking University, Beijing 102600, China; lxzhou@ss.pku.edu.cn
3 Institute of Geographic Sciences and Natural Resources Research, China Academy of Science, Beijing 100101, China
4 360 Security Technology Inc., Beijing 100015, China; liuqingjun@360.cn
* Correspondence: wangshaohua@lreis.ac.cn; Tel.: +86-010-59896521

Received: 13 January 2020; Accepted: 7 March 2020; Published: 30 March 2020

Abstract: Predicting the futures from previous spatiotemporal data remains a challenging topic. There have been many previous works on predictive learning. However, mainstream models suffer from huge memory usage or the gradient vanishing problem. Enlightened by the idea from the resnet, we propose CostNet, a novel recursive neural network (RNN)-based network, which has a horizontal and vertical cross-connection. The core of this network is a concise unit, named Horizon LSTM with a fast gradient transmission channel, which can extract spatial and temporal representations effectively to alleviate the gradient propagation difficulty. In the vertical direction outside of the unit, we add overpass connections from unit output to the bottom layer, which can capture the short-term dynamics to generate precise predictions. Our model achieves better prediction results on moving-mnist and radar datasets than the state-of-the-art models.

Keywords: spatiotemporal network; predictive learning; horizon LSTM; vertical structure; encoder-decoder architecture

1. Introduction

With the generation and preservation of big data, more spatiotemporal data are available in our daily life, which have the characteristics of spatial and temporal information [1–5]. Recently, spatiotemporal predictive learning has become a hot topic in practical applications [6], including precipitation nowcasting [7–9], crowd flows prediction [10–12], video prediction [13–17] and action recognition [18,19]. In contrast to traditional deep learning [20], predictive learning is able to predict future data from previous label-free spatiotemporal data in an unsupervised manner.

The vanishing gradient problem is a difficult research question when training artificial neural networks in predictive learning. It causes poor long-term prediction because the gradient based learning method will last much longer as errors vanish with back-propagation. Addressing the vanishing gradient problem is an issue in spatiotemporal predictive leaning.

There have been many previous studies on predictive learning, including recursive neural network (RNN) models, convolutional neural network (CNN) models, and generative adversarial network (GAN) methods. However, mainstream models suffer from huge memory usage or gradient vanishing problems [21–23]. Because predictive learning for spatiotemporal data, especially in the precipitation nowcasting, always deals with objects entangling, shape-changing, and direction variation, it is a more challenging task than the traditional temporal sequence regression, and it is a research direction worth exploring. The predictive learning based framework can solve the issue well, but the LSTM internal

unit structure is complicated. Our motivation is to explore a more straightforward unit structure and to solve or mitigate the gradient vanishing problem.

Toward a resolution of the gradient vanishing problem and simpler structure in the cell unit, we present CostNet, a novel RNN-based network. It is well known that resnet [1,24] excelled in the imagenet competition, which greatly increases the depth of convolution network by skip connections without causing gradient vanishing or gradient explosion problems. Enlightened by the idea of skip connections, CostNet has a horizontal and vertical cross connection. The core of this network is a concise unit, named Horizon LSTM with a fast gradient transmission channel, which provides a quick route from future predictions back to distant previous inputs to alleviate the gradient propagation difficulty. In the vertical direction outside of the unit, we add overpass connections from the unit output to the bottom layer, which can capture the short-term dynamics to generate clear predictions. Our model achieves better prediction results on moving-mnist and radar datasets than the state-of-the-art models, showing a great modeling capability for spatiotemporal data.

In this study, we propose a novel RNN-based network called CostNet. The paper is organized as follows. Related work is illustrated in Section 2. Section 3 introduces preliminaries. Section 4 shows the Horizon LSTM and vertical structure for the CostNet. Experiments and results are given in Section 5, and following is that is the Conclusion in Section 6.

2. Related Work

In recent years, a growing number of predictive learning models have been proposed, which are mainly based on convolutional neural network (CNN) [25], recursive neural network(RNN) [26,27] or generative adversarial network (GAN) [28,29].

Due to the powerful ability of extracting spatial correlations, CNN has achieved great success in the computer vision field, such as image classification and objects detection [20]. Some researchers attempted to model spatiotemporal data based on CNN. Oh et al. introduced an action autoencoder model based on CNN for video Atari games prediction [30], but its performance in real-world video is not good. De Brabandere et al. constructed the dynamic filter networks to some of input samples [31]. Zhang et al. designed the deep spatiotemporal networks for citywide crowd flows prediction using residual learning and fusion mechanism [12]. However, the model is only applied to very short-term prediction. Villegas et al. built a three-stage framework with additional annotated human joints data for long-term prediction [32]. However, it works in a supervision manner requiring a landmark as the ground truth.

Due to the powerful ability of modeling temporal dependencies, RNN has achieved great success in the natural language processing field, such as machine translation and intelligent conversational systems. Some researchers attempted to model spatiotemporal data based on RNN. Ranzato et al. introduced the first RNN framework inspired from language modeling and build a baseline for video prediction [14]. However, it has been shown that the model can only predict one frame ahead. Srivastava et al. employed the sequence to sequence LSTM network from language modeling to make multi-step video prediction [33]. The temporal characteristics are captured by the fully connected LSTM (FC-LSTM) layer in the model which cannot extract the spatial correlations. To learn spatial and temporal characteristics simultaneously, Shi et al. adopted convolution operator into input-to-state and state-to-state transitions and presented the convolutional LSTM (ConvLSTM) [7]. However, the stacked encoder-decoder architecture tends to produce fuzzy results. ConvLSTM becomes an important reference in the future research of spatiotemporal data because of its artful design. Finn et al. extended the convolutional LSTM model in robotics planning and constructed an action-conditioned video prediction network [34]. Patraucean et al. built a spatiotemporal video autoencoder with differentiable memory for action recognition [35], which can model short-term temporal dynamics and only predict one future frame partially related to optical flow and the convolutional LSTM. Villegas et al. also presented recurrent models based on the convolutional LSTM using optical flow as guided features to help capture short-term dynamics for video prediction and built an encoder-decoder

network that separates motion and content into different encoder pathways for pixel-level future prediction [36]. Lotter et al. proposed a deep predictive coding network upon ConvLSTM particularly designed for one-frame video prediction [15]. Shi et al. continued to explore a new model to solve the location-invariant problem and proposed a benchmark for precipitation nowcasting [8]. Combining gated CNN and ConvLSTM, Kalchbrenner et al. designed a sophisticated probabilistic video model, named Video Pixel Network (VPN) [16], which encodes a four-dimensional dependency chain from raw videos and estimates the discrete joint distribution of pixel values one-by-one. This model gives sharp prediction frames but also brings high computational complexity and low prediction efficiency. Unlike the stacked ConvLSTMs, Wang et al. proposed a novel encoder-decoder architecture (PredRNN) for spatiotemporal predictive learning adding zigzag memory flows from top layer to bottom layer which is beneficial for modeling short-term video dynamics and designed a complex unit [9], named ST-LSTM with dual-memory (temporal and spatiotemporal memory) flows as blocks in the network. Wang et al. continued to develop PredRNN++ with a unit GHU (Gradient Highway Unit) [37] to alleviate the deep-in-time dilemma and proposed a more reasonable but still complex unit, named causal LSTM.

Due to the powerful ability of generating similar patterns, GAN has become a hot research topic in the machine learning field, such as image style transfer and video generation. Some researchers attempted to model spatiotemporal data based on GAN. Mathieu et al. introduced generative adversarial networks to video prediction [17], which generate prediction frames by a generator and then distinguish real/fake frames by a discriminator. More methods about adversarial learning were present in video prediction [38–41]. These methods can generate sharper frames than the traditional CNN or RNN methods. However, they need careful training because of the unstable adversarial networks.

In summary, different approaches have different disadvantages. GAN-based approaches can generate sharp frames but not capture the temporal dynamics in the long-term prediction. Generally speaking, CNN-based approaches are also poor at long-term prediction because convolutional structures can extract the spatial correlations but not model the temporal dynamics effectively. On the contrary, RNN-based approaches are good at modeling temporal dependencies in the long-term prediction but tend to generate blur predictions because of the well-known vanishing gradient problem. In this study, we proposed a concise overpass spatiotemporal network, which can model spatial and temporal characteristics simultaneously.

3. Preliminaries

The goal of predictive learning for spatiotemporal data is to forecast future predictions using previous observation sequences. From a mathematical view, this task can be regarded as a probability estimation problem. We take a video clipping (a common format of spatiotemporal data) as a research object. It's a temporal sequence in general that spans from $t - J + 1$ to $t + K$. Given a time stamp t, $x_{t-J+1}, \ldots, x_t$ (length- J) represents the previous observations and $x_{t+1}, \ldots, x_{t+K}$ (length- K) represents the ground truth values of the future status. At the given time stamp t, each observation x, a spatial representation, can be represented by a tensor $R^{C \times M \times N}$, where R means the feature, C M and N denote the channel, height, width of a frame respectively. The essence of prediction is to predict the future length K sequence based on the known length J sequence and to maximize the prediction probability p. The predictions $\hat{x}_{t+1}, \ldots, \hat{x}_{t+k}$ are used as estimate values of the ground truth $x_{t+1}, \ldots, x_{t+K}$. This process can be implemented by an encoder-decoder architecture. Many models for predictive learning use the encoder-decoder architecture, including FC-LSTM, ConvLSTM, ST-LSTM, Cause LSTM and our model. First, the encoder is used to encode the previous observations into intermediate states, and

then the decoder is used to generate prediction results based on these intermediate states. The formulas are given in Formula (1) as follows:

$$\begin{aligned} \hat{x}_{t+1}, \ldots, \hat{x}_{t+k} &= \underset{x_{t+1}, \ldots, x_{t+K}}{\arg\max}\, p\left(x_{t+1}, \ldots, x_{t+K} \middle| x_{t-J+1}, \ldots, x_t\right) \\ &= \underset{x_{t+1}, \ldots, x_{t+K}}{\arg\max}\, p\left(x_{t+1}, \ldots, x_{t+K} \middle| f_{encoding}\left(x_{t-J+1}, \ldots, x_t\right)\right) \\ &= g_{\text{decoding}}\left(f_{encoding}\left(x_{t-J+1}, \ldots, x_t\right)\right) \end{aligned} \tag{1}$$

LSTM is suitable for processing temporal sequences, which is a recurrent cell unit with four gate structures inside. According to paper [27], the main formulas of LSTM are shown in Formula (2) below:

$$\begin{aligned} g_t &= \tan h\left(W_{xc} \bullet x_t + W_{hc} \bullet h_{t-1} + b_c\right) \\ i_t &= \sigma\left(W_{xi} \bullet x_t + W_{hi} \bullet h_{t-1} + W_{ci} \circ c_{t-1} + b_i\right) \\ f_t &= \sigma\left(W_{xf} \bullet x_t + W_{hf} \bullet h_{t-1} + W_{cf} \circ c_{t-1} + b_f\right) \\ c_t &= f_t \circ c_{t-1} + i_t \circ g_t \\ o_t &= \sigma\left(W_{xo} \bullet x_t + W_{ho} \bullet h_{t-1} + W_{co} \circ c_t + b_o\right) \\ h_t &= o_t \circ \tan h\left(c_t\right) \end{aligned} \tag{2}$$

where σ is sigmoid activation function, $\bullet$ and $\circ$ denote the matmul product and the Hadamard product respectively. However, for spatial data, the matmul product generates too many redundant connections (full connections) to extract efficiently spatial correlations with high efficiency.

By combining convolution layer and recursion layer, Shi et al. proposed ConvLSTM [7], which is widely used in the field of spatiotemporal data because spatial correlations and temporal dynamics are extracted simultaneously. ConvLSTM replace matmul product with convolution in full-connection LSTM cell. The main formulas of ConvLSTM are shown in Formula (3) below:

$$\begin{aligned} g_t &= \tan h\left(W_{xg} * x_t + W_{hg} * h_{t-1} + b_g\right) \\ i_t &= \sigma\left(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \circ c_{t-1} + b_i\right) \\ f_t &= \sigma\left(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \circ c_{t-1} + b_f\right) \\ c_t &= f_t \circ c_{t-1} + i_t \circ g_t \\ o_t &= \sigma\left(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \circ c_t + b_o\right) \\ h_t &= o_t \circ \tan h\left(c_t\right) \end{aligned} \tag{3}$$

where σ is sigmoid activation function, $*$ is the convolution operator and $\circ$ indicates the Hadamard product. However, the network only stacks four layers of ConvLSTM units vertically, independent of each other step-to-step, thus the bottom layer ignores characteristics extracted by the top layer at the previous time. Predictions cannot get the short-term trends and tend to be fuzzy.

To overcome the drawback of layer-independent architecture in ConvLSTM, Wang et al. proposed a novel encoder-decoder architecture (PredRNN) [9] with zigzag memory flows from the top layer to the bottom layer and designed a dual memory unit named ST-LSTM using complicated nonlinear transition functions. PredRNN has a strong capability of modelling short-term video dynamics and

generates clearer predictions than the ConvLSTM. The key equations of ST-LSTM are shown in (4) as follows:

$$\begin{aligned}
g_t &= \tanh\left(W_1 * \left[X_t, H_{t-1}^k, C_{t-1}^k\right]\right) \\
i_t &= \sigma\left(W_1 * \left[X_t, H_{t-1}^k, C_{t-1}^k\right]\right) \\
f_t &= \sigma\left(W_1 * \left[X_t, H_{t-1}^k, C_{t-1}^k\right]\right) \\
C_t^k &= f_t \circ C_{t-1}^k + i_t \circ g_t \\
g_t' &= \tanh\left(W_2 * \left[X_t, C_t^k, M_t^{k-1}\right]\right) \\
i_t' &= \sigma\left(W_2 * \left[X_t, C_t^k, M_t^{k-1}\right]\right) \\
f_t' &= \sigma\left(W_2 * \left[X_t, C_t^k, M_t^{k-1}\right]\right) \\
M_t^k &= f_t' \circ \tanh\left(W_3 * M_t^{k-1}\right) + i_t' \circ g_t' \\
o_t &= \tanh\left(W_4 * \left[X_t, C_t^k, M_t^k\right]\right) \\
H_t^k &= o_t \circ \tanh\left(W_5 * \left[C_t^k, M_t^k\right]\right)
\end{aligned} \tag{4}$$

where σ is sigmoid activation function, $*$ is the convolution operator and $\circ$ indicates the Hadamard product. The square brackets represent concatenation and the round brackets denote a whole section. Unfortunately, the values of gradient fall exponentially in the back-propagation process. The complicated ST-LSTM still suffers from the gradient vanishing problem [23].

4. Methodology

We present a novel method based on ST-LSTM to explore a straightforward unit structure and to mitigate the gradient vanishing problem. In this section, we will describe details of CostNet, a concise overpass spatiotemporal network. We adopt an encoder-decoder architecture with four layers and employ the Horizon LSTM as our backbone block. Our approach has two key insights: First, the core of this network is the Horizon LSTM, a concise unit with a fast gradient transmission channel, which can extract spatial and temporal representations effectively to alleviate the gradient propagation difficulty. Second, in the vertical direction outside of the unit, we add overpass connections from unit output to the bottom layer, which can capture the short-term dynamics to generate clear predictions.

4.1. Horizon LSTM

Similar to the ST-LSTM, our Horizon LSTM also has a dual-memory structure: the temporal memory C and the spatiotemporal memory M. The memory C that flows horizontally from previous step to next step captures the temporal dependencies. The memory M that moves vertically from the bottom layer to the top layer extracts the spatial correlations. The Horizon LSTM unit is enlightened by the idea of the skip connection from resnet instead of the complex gate structures from ST-LSTM. The structure of Horizon LSTM is shown in Figure 1. There are four inputs to Horizon LSTM, including X_t, H_{t-1}^l, C_{t-1}^l and M_t^{l-1}. X_t is the input frame in the first layer at the current time stamp. H_{t-1}^l is the output hidden states in the current layer at the previous time stamp. C_{t-1}^l is the temporal memory output states in the current layer at the previous time stamp. M_t^{l-1} is the spatiotemporal memory output states in the bottom layer at the current time stamp. When in the first layer, the input should be M_{t-1}, which is the spatiotemporal memory output states in the top layer at the previous time stamp. There are three outputs for Horizon LSTM: H_t^l, C_t^l and M_t^l. H_t^l is the output hidden states in the current layer at the current time stamp. C_t^l is the temporal memory output states in the current layer at the current time stamp. M_t^l is the spatiotemporal memory output states in the current layer at the current time stamp. Just like the ConvLSTM, the Horizon LSTM unit consists of input gate i_t, input modulation gate g_t, forget gate f_t and output gate o_t. The forget gate f_t controls the spatiotemporal information flow M. The temporal memory flow C depends the input gate i_t, the input modulation gate g_t and the forget gate f_t in our Horizon LSTM block. The output hidden states H_t^l in the current layer l and at the current time stamp t is determined by the temporal memory C_t^l as well as the output gate o_t. As shown in Figure 1, the spatiotemporal memory M exists in an overpass way through the gate structures in the Horizon LSTM just like the temporal memory C. Since there are only a few of blocks

in the memory route, the Horizon LSTM can provide a fast gradient transmission channel for both the temporal memory C and the spatiotemporal memory M from near predictions back to distant previous inputs to ease the gradient propagation difficulty.

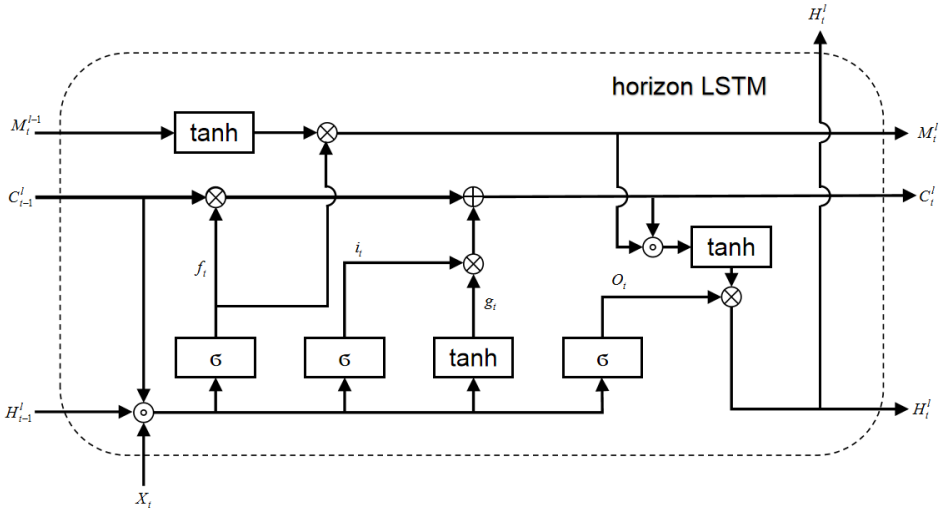

Figure 1. The architecture of Horizon LSTM, in which the temporal and spatiotemporal memories exist in overpass way through gate structures.

The key equations of the Horizon LSTM unit are shown in Formula (5) as follows:

$$\begin{aligned} g_t &= \tanh\left(W_1 * \left[X_t, H_{t-1}^l, C_{t-1}^l\right]\right) \\ i_t &= \sigma\left(W_1 * \left[X_t, H_{t-1}^l, C_{t-1}^l\right]\right) \\ f_t &= \sigma\left(W_1 * \left[X_t, H_{t-1}^l, C_{t-1}^l\right]\right) \\ C_t^l &= f_t \circ C_{t-1}^l + i_t \circ g_t \\ M_t^l &= f_t \circ \tanh\left(W_2 * M_t^{l-1}\right) \\ o_t &= \sigma\left(W_3 * \left[X_t, C_{t-1}^l, M_t^l\right]\right) \\ H_t^l &= o_t \circ \tanh\left(W_4 * \left[C_t^l, M_t^l\right]\right) \end{aligned} \tag{5}$$

where $*$ is the convolution operation, $\circ$ is the element-wise Hadamard product, σ is the Sigmoid function. The square brackets represent matrix concatenation and the round brackets denote a whole section. W 1~4 represent convolution filter parameters, where W 4 has a shape of 1×1 convolution filters to adjust the feature map output. All state variables can be represented by a four-dimensions tensor, which is comprised of the batch, width, height and hidden states. As illustrated in Formula 5, all of the input gate i_t, the input modulation gate g_t, the forget gate f_t and the output gate o_t are the functions of X_t, H_{t-1}^l, C_{t-1}^l. The temporal memory C_t^l is the function of the input gate i_t, the input modulation gate g_t, the forget gate f_t and the temporal memory output states C_{t-1}^l at the previous time stamp. The spatiotemporal memory M_t^l is the function of the forget gate f_t and the spatiotemporal memory output states M_t^{l-1} in the bottom layer. The output hidden states H_t^l is the function of the output gate o_t, the spatiotemporal memory M_t^l and the temporal memory C_t^l. Compared with equations (4) in ST-LSTM, our method has less gate structures and convolution operations shown in equations (5). The ST-LSTM has double input gate i_t, input modulation gate g_t, forget gate f_t, while our Horizon LSTM has only one input gate i_t, input modulation gate g_t, forget gate f_t. Therefore, our Horizon LSTM has a more concise structure than the ST-LSTM.

4.2. Vertical Structure

Similar to the encoder-decoder architecture of PredRNN, our CostNet also has a four-layers structure: The first and second layer form the encoder; The third and fourth layers form the decoder.

In the vertical direction, CostNet is enlightened by the idea of skip connection from resnet instead of the direct connection between the top layer and the bottom layer of PredRNN. Our network topology of our CostNet is illustrated in Figure 2. There are four rows and three columns in our CostNet, where rows denote different layers from the bottom to the top, columns denote different time stamp. H1 is short for the Horizon LSTM block in the first (bottom) layer while H4 is short for the Horizon LSTM block in the fourth (top) layer. $t-1$ indicates the previous time stamp while $t+1$ indicates the future time stamp. In fact, the block boxes at different time share the same parameters. X_t represents the input frame at the time stamp t while $\hat{X}_t$ represents the prediction result for X_t. The black arrows show the directions for the input frame or the output results. The orange arrows show the flow directions for the temporal memory C and the hidden states H while the blue arrows show the move directions for the spatiotemporal memory M and the hidden states H. It can be seen from the figure that temporal memory C only flows in the horizontal direction, the spatiotemporal memory M only moves in the vertical direction, while the hidden states H flows in both the horizontal direction and the vertical direction. The dotted lines represent the same implementation described by the solid lines. The symbol $\oplus$ means concatenation for tensors M. In the vertical direction outside of the unit, unlike the direct connection between the top layer and the bottom layer of PredRNN our CostNet places overpass connections from each unit output to the bottom layer. PredRNN only considers the influence of high-level semantics from the top layer of the prediction in the next step prediction, while CostNet considers the influence of different semantic features from each layer output, which can capture the short-term dynamic effectively. Therefore, the CostNet has more accurate and clearer predictions than that of PredRNN. Each layer of PredRNN extracts spatio-temporal features and the information flow flows from the bottom layer to the top layer, and the output of the last layer is fed back to the bottom layer. The CostNet can extract the spatio-temporal feature information each time, not only pass up but also pass down.

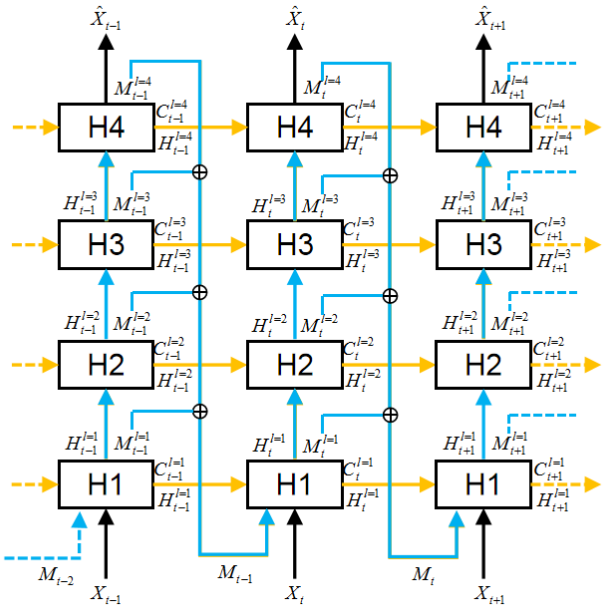

Figure 2. Overall architecture diagram.

The key equations of the entire CostNet are presented in Formula (6) as follows:

$$\begin{aligned} M_{t-1} &= \left[M_{t-1}^{l=1}, M_{t-1}^{l=2}, M_{t-1}^{l=3}, M_{t-1}^{l=4}\right] \\ H_t^{l=1}, C_t^{l=1}, M_t^{l=1} &= \text{HorizonLSTM}_1\left(X_t, M_{t-1}, H_{t-1}^{l=1}, C_{t-1}^{l=1}\right) \\ H_t^{l=2}, C_t^{l=2}, M_t^{l=2} &= \text{HorizonLSTM}_2\left(H_t^{l=1}, M_t^{l=1}, H_{t-1}^{l=2}, C_{t-1}^{l=2}\right) \\ H_t^{l=3}, C_t^{l=3}, M_t^{l=3} &= \text{HorizonLSTM}_3\left(H_t^{l=2}, M_t^{l=2}, H_{t-1}^{l=3}, C_{t-1}^{l=3}\right) \\ \hat{X}_t, C_t^{l=4}, M_t^{l=4} &= \text{HorizonLSTM}_4\left(H_t^{l=3}, M_t^{l=3}, H_{t-1}^{l=4}, C_{t-1}^{l=4}\right) \end{aligned} \tag{6}$$

HorizonLSTM_1 means the Horizon LSTM unit in the first (bottom) layer. Section on the left of the equal sign means the outputs for Horizon LSTM and sections in the round brackets represent the inputs for Horizon LSTM. The square brackets denote concatenation for the spatiotemporal memory M_{t-1} at the previous time stamp.

5. Experiments

In this section, we evaluate our model by comparing experiments on two datasets to demonstrate the effectiveness and advancement of our algorithm. At the beginning, we inform general configuration for our experiments. For each evaluation dataset, we introduce the dataset and implementation procedure. Then we show experimental results of our model as well as the baseline models. At last, we analyze performance quantitatively and visualize prediction examples qualitatively.

Our model was developed in python and implemented in Keras [42] with TensorFlow [43] as back-ends. All the experiments were run on the Ubuntu server with a single NVIDIA GTX1080Ti GPU. The general configures are listed as follows: (1) ADAM [44] optimizer is adopted with a starting learning rate of 10-3. (2) The batch size is set to 8. (3) The convolution filter size is set to 5 inside all recurrent units. (4) The objective function is L1 + L2 loss to generate sharp and smooth frames. (5) Maximum iterations is set to 6000. (6) Encoder and decoder structure consists of 4 LSTM layers with 128, 64, 64, 64 hidden state channels respectively. (7) Layer normalization [45] is adopted to avoid internal covariate shift problems. Besides, we employ the scheduled sampling strategy [46] to reduce differences between inference and training. In order to improve the training efficiency, we adopted the callback function in Keras, such as EarlyStopping, ModelCheckpoint and ReduceLROnPlateau. The source codes and data are available with a DOI at https://doi.org/10.6084/m9.figshare.11917914.v1.

5.1. Moving MNIST Dataset

5.1.1. Implementation

The Moving MNIST is a synthetic dataset constructed by moving digits from the MNIST dataset. It contains many data records, each of which is a sequence of length 20 (length of input frames is 10 and that of prediction frames is also 10. Each frame is a $64 \times 64 \times 1$ grayscale image including two handwritten digits bouncing inside. Because digits selection, initial position, velocity direction and velocity magnitude are random, it is difficult to predict future frame. We generate the sequences in the way introduced by Srivastava et al. [33]. We split the dataset volume is into the training set with 10,000 sequences, the validation set with 3000 sequences and the test set with 5000 sequences.

5.1.2. Results

The intuitive way to measure the uncertainty for predictive learning is variance. We adopted two quantitative metrics to evaluate the performance of all models. One metric is the mean square error (MSE), an objective indicator, which represents the distance between true frames and predictions. A better model should have a lower value of MSE. In the ideal case, the minimum value is zero. Another metric is the per-frame structural similarity index measure (SSIM) [47], which is a subjective indicator to measure similarity between two images. A better model should have a higher value of SSIM. In the ideal case, the maximum value is 1. Table 1 shows the performances of different models for predicting

10 frames given the previous 10 frames on the standard Moving MNIST dataset. As shown in the table, CostNet is evaluated against state-of the-art methods including FC-LSTM, ConvLSTM, TrajGRU, CDNA, DFN, FRNN, VPN, ST-LSTM and Causal LSTM. Our model outperforms all state-of-the-art methods in the metric MSE as well as SSIM. Our model reduces the per-frame MSE from 118.3 down to 44.9 and increases the per-frame SSIM from 0.690 up to 0.901. Compared with Causal LSTM, a recent state-of-the-art method, our model achieves competitive predictions, with a slight lower 1.6 in the metric MSE and a slight higher 0.03 in the metric SSIM. The results show that the CostNet can model spatiotemporal data effectively.

Table 1. A comparison of different models for predicting 10 frames on the Moving MNIST dataset.

Model	SSIM	MSE
FC-LSTM	0.690	118.3
ConvLSTM	0.707	103.3
TrajGRU	0.713	106.9
CDNA	0.721	97.4
DFN	0.726	89.0
FRNN	0.813	69.7
VPN	0.870	64.1
ST-LSTM	0.867	56.8
Causal LSTM	0.898	46.5
CostNet	0.901	44.9

We plot frame-wise curves of different models for predicting 10 frames. As illustrated in Figure 3, the CostNet is evaluated against state-of-the-art methods including FC-LSTM, ConvLSTM and Causal LSTM. Uniformly, the performance of all models declines over time. Nevertheless, our model outperforms the state-of-the-art methods, with a lower curve for the metric MSE and a higher curve for the metric SSIM. Compared with Causal LSTM, a recent state-of-the-art method, our model works slightly better, especially for the last four frames. The results indicate our model has a great power for capturing long-term video dependencies.

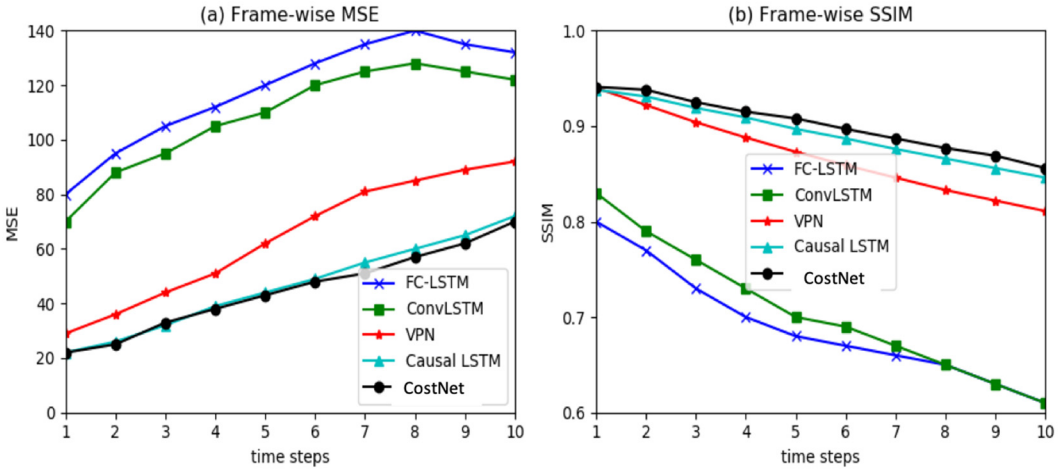

Figure 3. Frame-wise comparisons of different models for predicting 10 frames on the Moving MNIST.

At last, we visualize some examples on the Moving MNIST test set to observe the performance of different models qualitatively. All models predict 10 frames in the future given 10 previous frames. As illustrated in Figure 4, the first row is the previous input frames, the second row is the ground truth data, the third to eleventh rows are the predictions of FC-LSTM, ConvLSTM, TrajGRU, CDNA, DFN, FRNN, VPN, ST-LSTM, Causal LSTM respectively, and the last row is the predictions of our model. We observe that our model predictions are sharp enough.

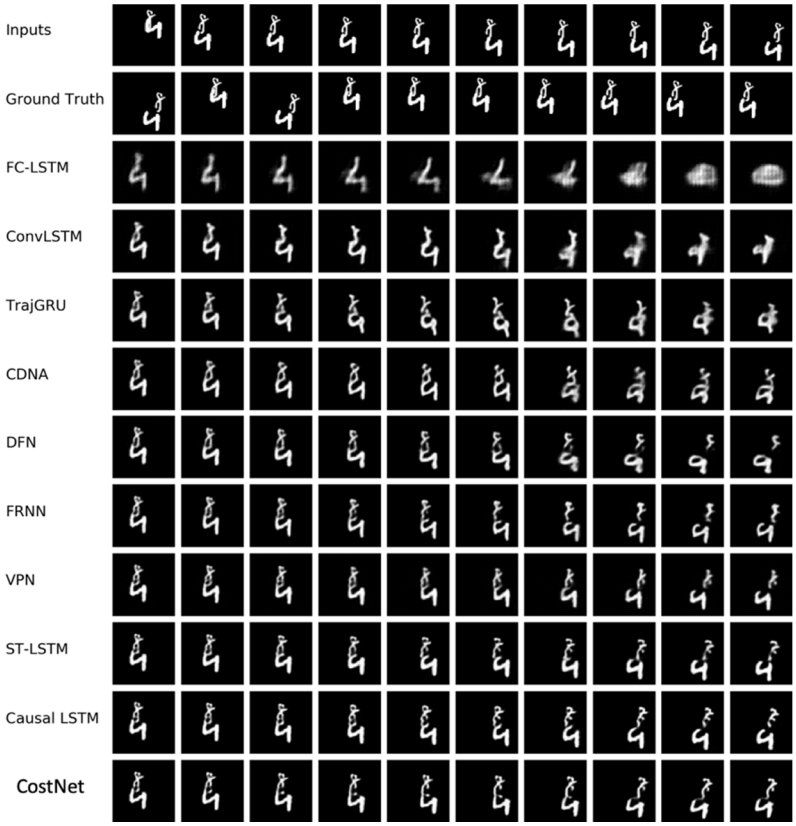

Figure 4. Prediction examples on the Moving MNIST test.

5.2. Radar Echo Dataset

5.2.1. Implementation

In order to verify the effectiveness and advancement of our mode, a practical dataset, standard radar dataset 2018 (SRAD2018), is adopted in experiment, which comes from the IEEE ICDM 2018 global weather AI challenge. The radar dataset spans four months from 00:00 UTC on March 15 to 23:54 UTC on July 15 each year from 2010 to 2017. There are 320,000 sets in this dataset, including 300,000 records as training set and 20,000 records as test set. Each record has a length of 61, with an interval of 6 min. Radar covers one vertical level, altitude 3 km. After quality control, the radar echo data is limited in 0-80 (unit: dBZ), and the missing value is 255. The radar data at each time is stored in grayscale PNG format with a resolution of 501 × 501. Our goal in experiment is to predict the future 10 frames based on the previous 10 consecutive frames. We did some data preprocessing, such as image reshaping to 200 × 200. In addition, we under sampled the original dataset taking one image at every three intervals. After preprocessing, the training set is 80,000 sequences, the verification set is 10,000 sequences, and the test dataset is 1000 sequences.

5.2.2. Results

We adopted three quantitative metrics to evaluate the performance of all models, including the mean square error (MSE), the per-frame structural similarity index measure (SSIM) and the per-frame Peak Signal to Noise Ratio (PSNR) [47]. SSIM focuses on the difference in sharpness while PSNR emphasizes pixel-level correctness. A better model should have a higher value of SSIM and PSNR. In the ideal case, the maximum value of SSIM is 1 and the maximum value of PSNR is 255. Table 2 shows the performance of different models for predicting 10 frames given the previous 10 frames on the radar dataset. As shown in the table, our model is evaluated against state-of-the-art methods including ConvLSTM, TrajGRU, ST-LSTM and Causal LSTM. Our model outperforms all state-of-the-art methods

in the metric SSIM as well as PSNR. Our model reduces the per-frame MSE from 3580.31 down to 888.81, increases the per-frame SSIM from 0.62 up to 0.79 and increases the per-frame PSNR from 12.13 up to 17.48. Compared with Causal LSTM, a recent state-of-the-art method, our model achieves competitive predictions, with a slightly higher 0.14 in the metric PSNR and 0.01 in the metric SSIM. The results show that CostNet can model radar data effectively.

Table 2. A comparison of different models for predicting 10 frames on the radar dataset.

Model	MSE	SSIM	PSNR
ConvLSTM	3580.31	0.62	12.13
TrajGRU	2088.88	0.69	14.13
ST-LSTM	1252.49	0.74	15.9
Causal LSTM	905.16	0.78	17.34
CostNet	888.81	0.79	17.48

We plot frame-wise curves of different models for predicting 10 frames. Better predictions should have higher curves of frame-wise SSIM and PSNR. As illustrated in Figure 5, our model is evaluated against state-of-the-art methods including ConvLSTM, TrajGRU, ST-LSTM and Causal LSTM. Uniformly, the performance of all models declines over time. Nevertheless, our model outperforms the state-of-the-art methods, with higher curves for the metric SSIM and PSNR. Compared with Causal LSTM, a recent state-of-the-art method, our model works slightly better, especially for the last four frames. The results indicate our model has a great power for capturing long-term video dependencies. The performance of CostNet for the SSIM has improved by 0.1 than that in the Causal LSTM. The significant improvement is mainly in the final 5 frames. The result showed that CostNet has a stronger ability in predicting long-term temporal scenarios.

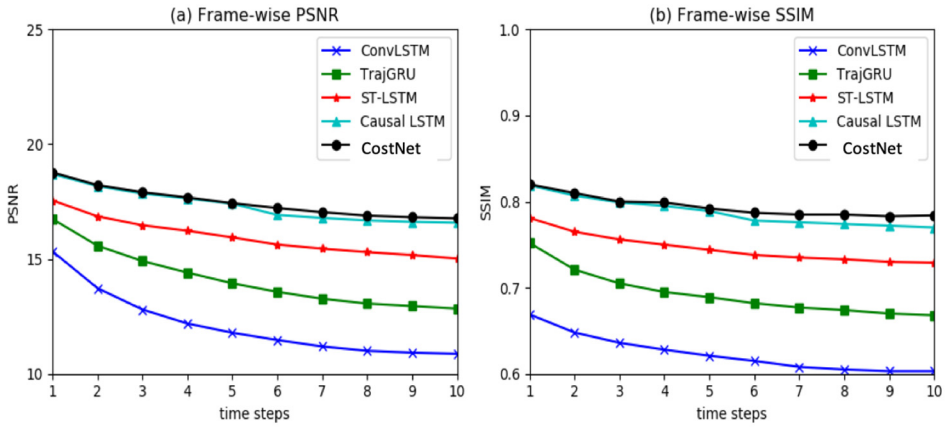

Figure 5. Frame-wise comparisons of different models for predicting 10 frames on the radar dataset.

We visualize examples on the radar test set to observe the performance of different models qualitatively. All models predict 10 frames in the future given 10 previous frames. As illustrated in Figure 6, the first row is the previous input frames, the second row is the ground truth, the third to sixth rows are the predictions of ConvLSTM, TrajGRU, ST-LSTM and Causal LSTM respectively, and the last row is the predictions of our model. We observe that our model predictions are sharp enough.

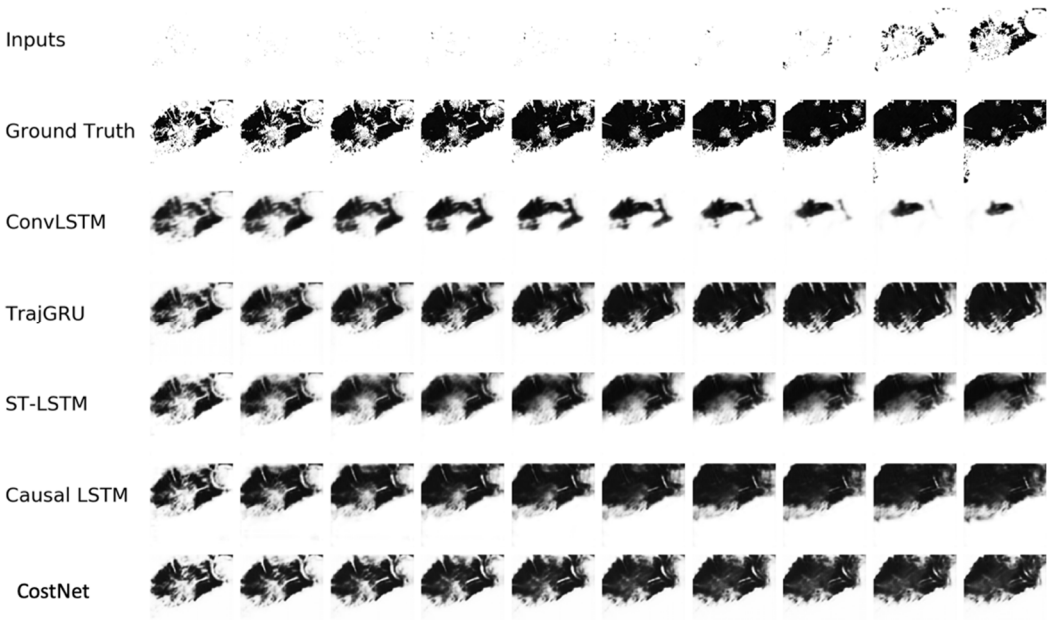

Figure 6. Prediction examples on the radar test set.

6. Conclusions and Future Work

Towards resolutions of gradient vanishing problem and simpler structure in the cell unit, we propose CostNet, a concise overpass spatiotemporal network, which has a horizontal and vertical cross connection. The core of the CostNet is a concise unit, named Horizon LSTM with a fast gradient transmission channel, which provide a quick route from near predictions back to distant previous inputs to ease the gradient propagation difficulty. In the vertical direction outside of the unit, we add overpass connections from unit output to the bottom layer, which can capture the short-term dynamics to generate clearer predictions. The CostNet can extract the spatio-temporal feature information each time. The CostNet achieves better prediction results on moving-mnist and radar datasets than the state-of-the-art models. The results showed that the CostNet has a great modeling capability for spatiotemporal data. However, the evaluation for the uncertainty of the predictive learning is only considering the MSE, SSIM and PSNR. The Critical success index (CSI) can be used for the evaluation the uncertainty of the dynamic predictive learning as a skill score index [48]. A higher CSI denotes a better prediction. We will evaluate the uncertainty of the predictive learning results using CSI in the future. The predictive learning for spatiotemporal data is still an extremely challenging topic for multiple predictive learning methods. In the near future, we will explore different network structures and predictive learning working in an unsupervised manner over cyberinfrastructure.

Author Contributions: Conceptualization, Fengzhen Sun, Shaojie Li, and Shaohua Wang; methodology, Fengzhen Sun, Shaojie Li, and Shaohua Wang; software, Fengzhen Sun and Shaojie Li; validation, Fengzhen Sun, Shaojie Li, and Shaohua Wang; formal analysis, Fengzhen Sun, Shaojie Li, and Shaohua Wang; investigation, Fengzhen Sun, Shaojie Li, and Shaohua Wang; resources, Fengzhen Sun, Shaojie Li, Shaohua Wang, Qingjun Liu, and Lixin Zhou; data curation, Fengzhen Sun, Shaojie Li, Shaohua Wang, Qingjun Liu, and Lixin Zhou; writing—original draft preparation, Fengzhen Sun, Shaojie Li, and Shaohua Wang; writing—review and editing, Fengzhen Sun, Shaojie Li, Shaohua Wang, Qingjun Liu, and Lixin Zhou; visualization, Fengzhen Sun, Shaojie Li, Shaohua Wang, Qingjun Liu, and Lixin Zhou; supervision, Shaojie Li; project administration, Fengzhen Sun and Shaojie Li. All authors have read and agreed to the published version of the manuscript.

Funding: Our work was supported by the National Key R&D Plan (2016YFB0502000), Project of Beijing Excellent Talents (201500002685XG242), National Postdoctoral International Exchange Program (Grant No. 20150081).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, M.; Zhang, X.; Niu, X.; Wang, F.; Zhang, X. Scene classification of high-resolution remotely sensed image based on resnet. *J. Geov. Spatial Anal.* **2019**, *3*, 16. [CrossRef]
2. Wang, S.; Zhong, Y.; Wang, E. An integrated GIS platform architecture for spatiotemporal big data. *Future Gener. Comput. Syst.* **2019**, *94*, 160–172. [CrossRef]
3. Liu, K.; Gao, S.; Qiu, P.; Liu, X.; Yan, B.; Lu, F. Road2vec: Measuring traffic interactions in urban road system from massive travel routes. *ISPRS Int. J. Geo Inf.* **2017**, *6*, 321. [CrossRef]
4. Liu, Y.; Cao, G.; Zhao, N. Integrate machine learning and geostatistics for high-resolution mapping of ground-level pm2. 5 concentrations. In *Spatiotemporal Analysis of Air Pollution and Its Application in Public Health*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 135–151.
5. Li, H.; Liu, J.; Zhou, X. Intelligent map reader: A framework for topographic map understanding with deep learning and gazetteer. *IEEE Access* **2018**, *6*, 25363–25376. [CrossRef]
6. LeCun, Y. Predictive learning. *Proc. Speech NIPS* **2016**. Available online: https://drive.google.com/file/d/0BxKBnD5y2M8NREZod0tVdW5FLTQ/view (accessed on 12 March 2020).
7. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; Woo, W.-C. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 12–17 December 2015; pp. 802–810.
8. Shi, X.; Gao, Z.; Lausen, L.; Wang, H.; Yeung, D.-Y.; Wong, W.-k.; Woo, W.-C. Deep learning for precipitation nowcasting: A benchmark and a new model. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5617–5627.
9. Wang, Y.; Long, M.; Wang, J.; Gao, Z.; Philip, S.Y. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 879–888.
10. Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X. DNN-based prediction model for spatio-temporal data. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Burlingame, CA, USA, 31 October–3 November 2016; ACM: Burlingame, CA, USA, 2016; pp. 1–4.
11. Xu, Z.; Wang, Y.; Long, M.; Wang, J.; KLiss, M. PredCNN: Predictive learning with cascade convolutions. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018; pp. 2940–2947.
12. Zhang, J.; Zheng, Y.; Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–10 February 2017.
13. Oliu, M.; Selva, J.; Escalera, S. Folded recurrent neural networks for future video prediction. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 716–731.
14. Ranzato, M.; Szlam, A.; Bruna, J.; Mathieu, M.; Collobert, R.; Chopra, S. Video (language) modeling: A baseline for generative models of natural videos. *arXiv* **2014**, arXiv:1412.6604.
15. Lotter, W.; Kreiman, G.; Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv* **2016**, arXiv:1605.08104.
16. Kalchbrenner, N.; van den Oord, A.; Simonyan, K.; Danihelka, I.; Vinyals, O.; Graves, A.; Kavukcuoglu, K. Video pixel networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*; JMLR: Sydney, Australia, 2017; pp. 1771–1779.
17. Mathieu, M.; Couprie, C.; LeCun, Y. Deep multi-scale video prediction beyond mean square error. *arXiv* **2015**, arXiv:1511.05440.
18. Jain, A.; Zamir, A.R.; Savarese, S.; Saxena, A. Structural-rnn: Deep learning on spatio-temporal graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5308–5317.
19. Tran, D.; Bourdev, L.D.; Fergus, R.; Torresani, L.; Paluri, M. C3D: Generic features for video analysis. *CoRR abs/1412.0767* **2014**, *2*, 8.
20. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
21. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]

22. Williams, R.J.; Zipser, D. Gradient-based learning algorithms for recurrent. In *Backpropagation: Theory, Architectures, and Applications*; Psychology Press: Brighton, UK, 1995; Volume 433.
23. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International conference on machine learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
25. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
26. Jordan, M.I. Serial order: A parallel distributed processing approach. In *Advances in Psychology*; Elsevier: Amsterdam, The Netherlands, 1997; Volume 121, pp. 471–495.
27. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
28. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
29. Denton, E.L.; Chintala, S.; Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; pp. 1486–1494.
30. Oh, J.; Guo, X.; Lee, H.; Lewis, R.L.; Singh, S. Action-conditional video prediction using deep networks in atari games. In Proceedings of the Advances in neural information processing systems, Montréal, QC, Canada, 7–12 December 2015; pp. 2863–2871.
31. Jia, X.; De Brabandere, B.; Tuytelaars, T.; Gool, L.V. Dynamic filter networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona Spain, 5–10 December 2016; pp. 667–675.
32. Villegas, R.; Yang, J.; Zou, Y.; Sohn, S.; Lin, X.; Lee, H. Learning to generate long-term future via hierarchical prediction. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*; JMLR: Sydney, Australia, 2017; pp. 3560–3569.
33. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised learning of video representations using LSTMs. In Proceedings of the International conference on machine learning, Lille, France, 6–11 July 2015; pp. 843–852.
34. Finn, C.; Goodfellow, I.; Levine, S. Unsupervised learning for physical interaction through video prediction. In Proceedings of the Advances in neural information processing systems, Barcelona, Spain, 5–10 December 2016; pp. 64–72.
35. Patraucean, V.; Handa, A.; Cipolla, R. Spatio-temporal video autoencoder with differentiable memory. *arXiv* **2015**, arXiv:1511.06309.
36. Villegas, R.; Yang, J.; Hong, S.; Lin, X.; Lee, H. Decomposing motion and content for natural video sequence prediction. *arXiv* **2017**, arXiv:1706.08033.
37. Wang, Y.; Gao, Z.; Long, M.; Wang, J.; Yu, P.S. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. *arXiv* **2018**, arXiv:1804.06300.
38. Vondrick, C.; Pirsiavash, H.; Torralba, A. Generating videos with scene dynamics. In Proceedings of the Advances In Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 613–621.
39. Lu, C.; Hirsch, M.; Scholkopf, B. Flexible spatio-temporal networks for video prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6523–6531.
40. Denton, E.L. Unsupervised learning of disentangled representations from video. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4414–4423.
41. Bhattacharjee, P.; Das, S. Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4268–4277.
42. Chollet, F. Keras: The python deep learning library. Available online: https://keras.io/#support (accessed on 12 March 2020).

43. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M. Tensorflow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv* **2016**, arXiv:1603.04467.
44. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
45. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
46. Bengio, S.; Vinyals, O.; Jaitly, N.; Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 12–17 December 2015; pp. 1171–1179.
47. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
48. Wang, Y.; Zhang, J.; Zhu, H.; Long, M.; Wang, J.; Yu, P.S. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9154–9162.