MODERNN: TOWARDS FINE-GRAINED MOTION DETAILS FOR SPATIOTEMPORAL PREDICTIVE LEARNING

Zenghao Chai[1]   *Zhengzhuo Xu*[1]   *Chun Yuan*[1,2*]

[1] Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2] Peng Cheng Laboratory, Shenzhen, China
zenghaochai@gmail.com; xzz20@mails.tsinghua.edu.cn; yuanc@sz.tsinghua.edu.cn

ABSTRACT

Spatiotemporal predictive learning (ST-PL) aims at predicting the subsequent frames via limited observed sequences, and it has broad applications in the real world. However, learning representative spatiotemporal features for prediction is challenging. Moreover, chaotic uncertainty among consecutive frames exacerbates the difficulty in long-term prediction. This paper concentrates on improving prediction quality by enhancing the correspondence between the previous context and the current state. We carefully design Detail Context Block (DCB) to extract fine-grained details and improve the isolated correlation between upper context state and current input state. We integrate DCB with standard ConvLSTM and introduce Motion Details RNN (MoDeRNN) to capture fine-grained spatiotemporal features and improve the expression of latent states of RNNs to achieve significant quality. Experiments on Moving MNIST and Typhoon datasets demonstrate the effectiveness of the proposed method. MoDeRNN outperforms existing state-of-the-art techniques qualitatively and quantitatively with lower computation loads.

Index Terms— Spatiotemporal prediction, Recurrent neural network, MoDeRNN, fine-grained details

1. INTRODUCTION

Spatio-**T**emporal **P**redictive **L**earning (ST-PL) is challenging with broad applications in predictive learning, e.g., physical object movement [1, 2, 3, 4], meteorological prediction [5, 6, 7, 8]. It aims to predict future sequences based on limited observed frames. The difficulty of ST-PL lies in the chaotic motion trends and profound dynamic changes. Hence it's necessary and crucial to build a proper corresponding between current input frames and previous observations, and integrate the motion trends for subsequent prediction.

Recent years have achieved impressive progress in ST-PL, plenty of novel approaches [5, 9, 10, 11, 12, 7, 13, 14] are proposed for long-term prediction. As one of the most popular branches, RNN [15] or LSTM [16] plays an important role

as a mainstream model. These methods have shown impressive results in ST-PL and made persistent progress. RNN and LSTM based models predict subsequent frames in an auto-regressive mode, i.e., stacked RNN layers embed input features obtained by CNN layers into latent states and update hidden states by the elaborate designed process to obtain output states, and decode to obtain the next timestamp frames.

However, when rethinking the calculation process of ConvLSTM [5, 9] and its extensions [10, 12, 7, 14], it's intuitive that the input state and upper context state show isolated correspondence in the process of RNN layers. The two states in previous models are only correlated by CNN layers and channel-wise addition operation. Hence models confront the two severe dilemmas that will lead to worse prediction results in long-term prediction: 1. The increasing models' depth and complexity exacerbate the declination of correlations between the current input and upper context, making it even difficult to build correct correspondence between the current frame and upper context. 2. CNNs can hardly capture fine-grained features that contain abundant details for prediction, limiting the ability to consider detailed features of latent states.

On top of the aforementioned, the current frame states are highly correlated to its neighbors of specific regions, i.e., the next timestamp frames in a region are related to both itself and its neighbor subject movements. The fine-grained local information is crucial for long-term prediction. To improve the correlation and the detailed local information between input and context, we propose Motion Details RNN (MoDeRNN) to tackle the above challenges in ST-PL effectively.

MoDeRNN contains the carefully designed Detail Context Block (DCB), which weights input and context states to highlight the spatiotemporal details for subsequent prediction. In specific, to obtain latent spatiotemporal trends among different neighbors, DCB utilizes various perceptual fields CNN layers to capture regions corresponding to input states and context states, and updates the corresponding context state and input state iteratively with rich correlations. As a result, the proposed MoDeRNN enables to capture fine-grained locals to persist correlations among RNN layers and achieves remarkable satisfactory prediction performance.
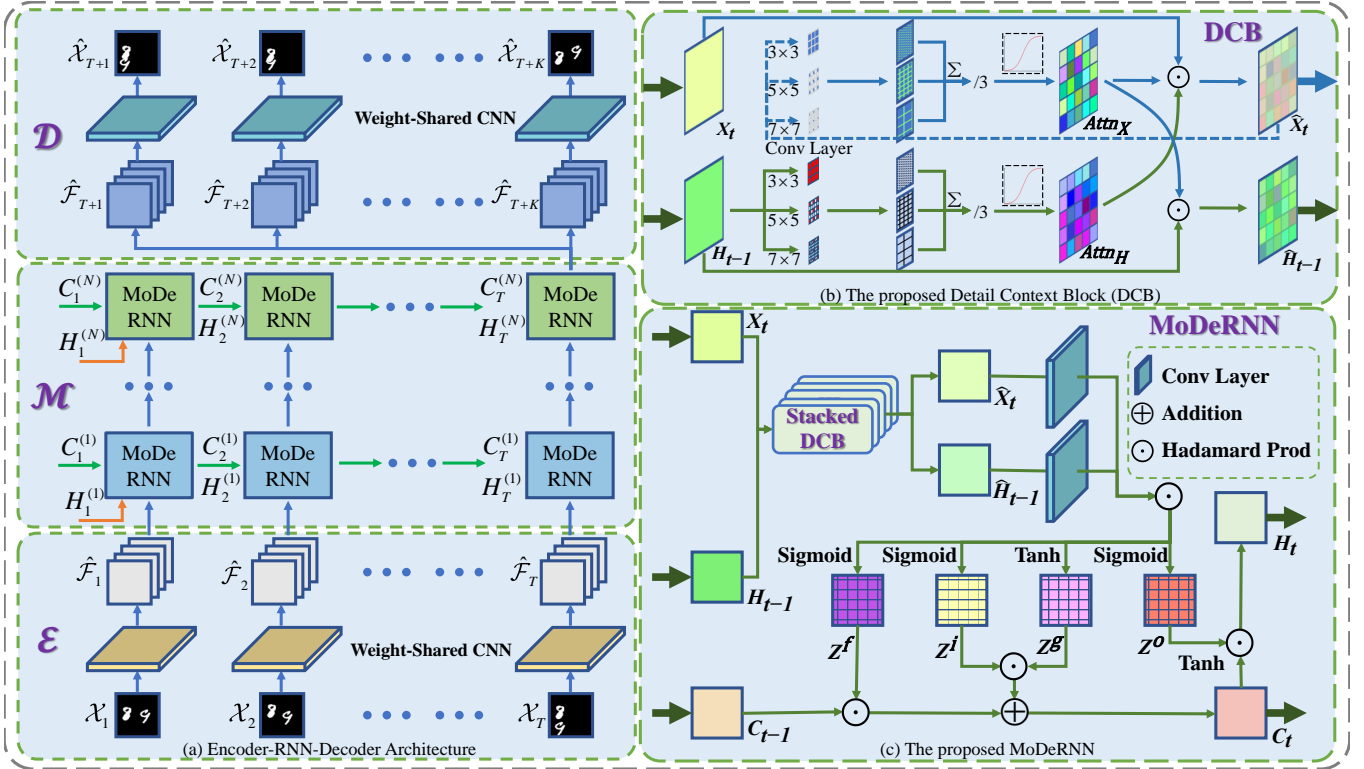
---

*Corresponding Author

Fig. 1: Overview of the proposed Method. (a): mainstream architecture for ST-PL; (b): pipeline of DCB; (c): pipeline of the proposed MoDeRNN.

In summary, our main contributions are two-fold:

- We construct Detail Context Block to capture fine-grained local details and update context states with dense correlations. We analyze the essential of context attention over fine-grained regions for prediction, and propose MoDeRNN towards fined-grained detailed prediction quality.

- We validate that the proposed MoDeRNN well captures trend regions of given frames to obtain better context and input correlations, and achieves distinguish performance gains compared to previous methods with fewer params on two representative datasets.

2. METHODOLOGY

2.1. Model Architecture

The RNN-based models are universally used approaches for ST-PL, with common Encoder-RNN-Decoder architecture [5, 10, 12, 7, 14] as Fig.1(a) shows. The given frames are encoded by 2D CNN [17] encoder $\mathcal{E}$ in step-by-step mode, then the obtained features $[\hat{\mathcal{F}}_{1:T}]$ serve as the input of N-layer LSTMs denoted as $\mathcal{M}$ to generate high-order spatiotemporal features of given sequences and output states $[\hat{\mathcal{F}}_{T+1:T+K}]$. Ultimately, the output states are decoded by 2D CNN decoder $\mathcal{D}$ iteratively and thus generate the next K frames $[\hat{\mathcal{X}}_{T+1:T+K}]$. The mathematics pipeline is illustrated as Eq.1.

$$
\begin{aligned}
[\hat{\mathcal{X}}_{1:T}] &= \mathcal{E}([\mathcal{X}_{1:T}]) \\
[\hat{\mathcal{F}}_{T+1:T+K}] &= \mathcal{M}([\hat{\mathcal{X}}_{1:T}]) \\
[\hat{\mathcal{X}}_{T+1:T+K}] &= \mathcal{D}([\hat{\mathcal{F}}_{T+1:T+K}])
\end{aligned}
\tag{1}
$$

In this paper, we keep the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ consistent with previous work mentioned above([10, 14], etc.). Namely, they are both 1×1 kernel CNN layers. The crucial target is to make representative high-order spatiotemporal features on RNNs, while there are issues worthy of consideration.

2.2. The proposed DCB

The abundant context feature is hard to obtain due to the limited operation between the current input and the previous context state in RNNs. In naïve ConvLSTM, the correlation of input state and context state is operated by CNN layers and add operation, while the subsequent state updates don't involve the interaction of the two states. Therefore, their relationship remains independent in the following operation, which easily leads to the loss of information in the prediction results. Intuitively, the output frames will get increasingly worse prediction quality, especially in detail parts.

Considering the importance of improving the correlation between neighbors and context states, we propose Detail Context Block (DCB) to extract fine-grained local features of cur-

rent input state X_t and context state H_{t-1}, and utilize the proposed context interaction approach to improve the correlations between the upper context state and current input state. The detailed architecture of DCB is illustrated in Fig.1(b).

In specific, to extract fine-grained local features, we utilize CNN layers with different perceptual fields to comprehensively focus on detailed motion regions of context state and input state, respectively, and use iterative weight correlate operations to improve the isolated correlation between the two states. DCB consists of the following steps:

Step 1. To obtain specific regions in current input state X_t that are essential for prediction, we comprehensively consider the influence of locals by generating an attention weight map $Attn_H$ of upper context via multi-kernel CNN layers that capture the context features, then obtain the mean local features that indicates the potential movement trend in the following timestamp. We adopt $Sigmoid$ function σ to normalize the weight map into $(0, 1)$, and reweight the input feature X_t by the Hadamard product to highlight the important part of the input state. Finally, we multiply the weight map by a constant scale factor s to avoid getting increasingly smaller.

Step 2. We encourage upper context H_{t-1} updates by considering the trends of current input, i.e., enforce H_{t-1} enlighten the fine-grained motion details and weaken the negligible parts with lower expression simultaneously. We update H_{t-1} by multiplying an input-related attention weight map $Attn_X$ to extract motion concentration for prediction by Hadamard product. The weight map is calculated the same as Step 1, i.e., capture the detailed context motion features by multi-kernel size CNN layers and activation function σ with scale factor s. Then, the updated context state $\hat{H}_{t-1}$ and input state $\hat{X}_t$ are obtained with rich spatiotemporal features.

2.3. Overview of MoDeRNN

Considering improving the expression ability in detail regions for ST-PL, we integrate DCB with naïve ConvLSTM to compose the proposed MoDeRNN as Fig.1(c), a new spatiotemporal prediction model towards fine-grained details. Formally, MoDeRNN can be expressed as follows:

Firstly, we utilize DCB to capture fine-grained detail spatiotemporal features and update current input state X_t and upper context state H_{t-1}. Then, to further improve the correlations between the two states, we use m stacked DCB with kernel size varies from $k \in \{3, 5, 7\}$ to improve the expression ability of MoDeRNN with more details.

$$
\begin{aligned}
Attn_H &= \sigma\left(\sum_i^k W_h^{i\times i} \star H_{t-1}/|k|\right) \\
\hat{X}_t &= s \times Attn_H \times H_{t-1} \\
Attn_X &= \sigma\left(\sum_i^k W_x^{i\times i} \star \hat{X}_{t-1}/|k|\right) \\
\hat{H}_{t-1} &= s \times Attn_X \times \hat{X}_{t-1}
\end{aligned}
\tag{2}
$$

where $W_h^{i\times i}$ and $W_x^{i\times i}$ represent $i \times i$ kernel CNN layers for H_{t-1} and $\hat{X}_t$, respectively. s represents the scale factor and σ indicates $Sigmoid$ activation function.

Secondly, we utilize the updated $\hat{X}_t$ and $\hat{H}_{t-1}$ to obtain the detailed output state H_t and memory state C_t. In the last layer of MoDeRNN, the final output state H_t is decoded to generate the final output frame of the next timestamp.

$$
\begin{aligned}
g_t &= \tanh(W_{xg} \star \hat{X}_t + W_{hg} \star \hat{H}_{t-1} + b_g) \\
i_t &= \sigma(W_{xi} \star \hat{X}_t + W_{hi} \star \hat{H}_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} \star \hat{X}_t + W_{hf} \star \hat{H}_{t-1} + b_f) \\
C_t &= f_t \circ C_{t-1} + i_t \circ g_t \\
o_t &= \sigma(W_{xo} \star \hat{X}_t + W_{ho} \star \hat{H}_{t-1} + b_o) \\
H_t &= o_t \circ \tanh(C_t)
\end{aligned}
\tag{3}
$$

where $W_{xg}, W_{hg}, W_{xi}, W_{hi}, W_{xf}, W_{hf}$ are 5×5 kernel CNN layers for gate operation.

3. EXPERIMENT

3.1. Experiment Details

We implement the proposed model by Pytorch [18], train and test it on a single RTX 2080Ti. For fair comparisons, we use 4-layer MoDeRNN units with 64-dim hidden states consistent with previous work. We set the mini-batch as 32 with the initial learning rate as 0.001. We also adopt scheduled sampling [19] and layer normalization [20] for better results. During training, we use $L_1 + L_2$ loss with AdamW [21] optimizer. Code is avaiable at https://github.com/czh-98/MoDeRNN.

3.2. Dataset

Moving MNIST. Moving MNIST [22] is a widespread benchmark for depicting 2 digits' movement with constant velocity. It contains $64 \times 64 \times 1$ consecutive frames with 10 for input and 10 for prediction, $10,000$ randomly generated sequences for training and $10,000$ fixed parts for testing.
Typhoon. Typhoon dataset is a meteorology radar data released by CEReS [23]. We resize the images into $64 \times 64 \times 1$ resolution and normalize to $[0, 1]$, then split the generated sequences into train and test sets. We use the given 8-hour observation data to predict the next 4 hours, with $1,809$ sequences for training and 603 sequences for testing.

3.3. Comparisons on Moving MNIST

We set $80,000$ iterations consistent with previous work ([10] etc.). We use PSNR, SSIM, MSE, and MAE for quantitative comparisons. The higher SSIM / PSNR and lower MSE / MAE indicate better performance. Results in Tab.1 demonstrate the superiority of our method on Moving MNIST dataset in all above metrics, improving **9.62**% and **2.52**% on PSNR and SSIM, and reducing **12.03**% and **27.46**% on MSE and MAE respectively compared with SA-ConvLSTM
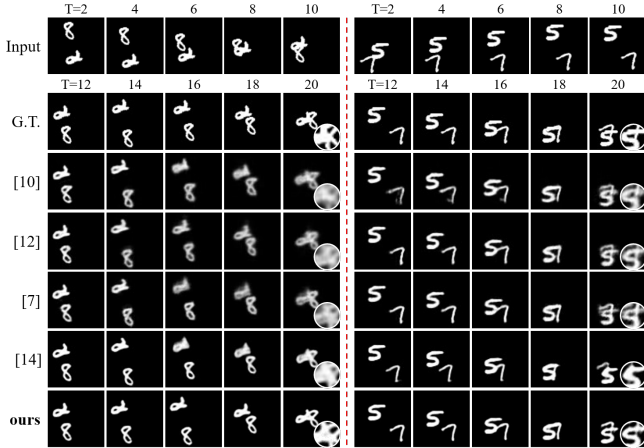
Fig. 2: Qualitative comparisons of previous SOTA models on Moving MNIST test set at $80,000$ iterations.

Models	# Params	PSNR ↑	SSIM ↑	MSE ↓	MAE ↓
DDPAE [24]	-	21.170	0.922	38.9	90.7
CrevNet [25]	-	-	0.928	38.5	-
PDE-Driven [26]	-	21.760	0.909	-	-
PredRNN [10]	13.799 M	19.603	0.867	56.8	126.1
PredRNN++ [12]	13.237 M	20.239	0.898	46.5	106.8
MIM* [7]	27.971 M	20.678	0.910	44.2	101.1
E3D-LSTM [13]	38.696 M	20.590	0.910	41.7	87.2
SA-ConvLSTM [14]	10.471 M	20.500	0.913	43.9	94.7
MoDeRNN (ours)	**4.590 M**	**22.472**	**0.936**	**30.6**	**68.7**

Table 1: Quantitative comparisons of previous SOTA models on Moving MNIST test set. All models predict 10 frames by observing 10 previous frames.

[14], while achieving lower computational loads. Fig.2 shows that MoDeRNN well preserves the variation details over digits, especially deals with the trajectory of overlaps and maintains the clarity over time. In contrast, other methods confront severe blurry challenges and are incapable of dealing with overlapped digits.

Fig.3 illustrates the weight map over consecutive timestamps. MoDeRNN focuses on fine-grained local details for subsequent prediction and can even handle overlap scenarios.

3.4. Comparisons on Typhoon

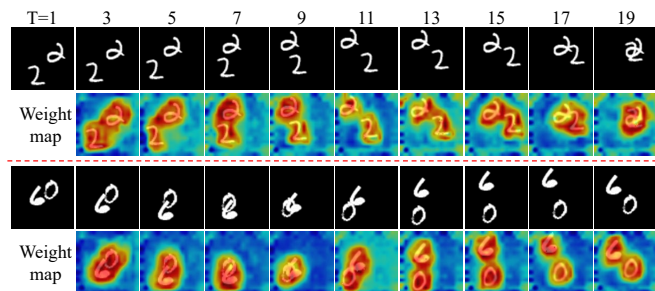We train the proposed models for $100,000$ iterations and make fair comparisons with previous methods [5, 10, 12, 7,

Fig. 3: Visualization of MoDeRNN on Moving MNIST test set of the last layer, the warm colors indicate higher weights.
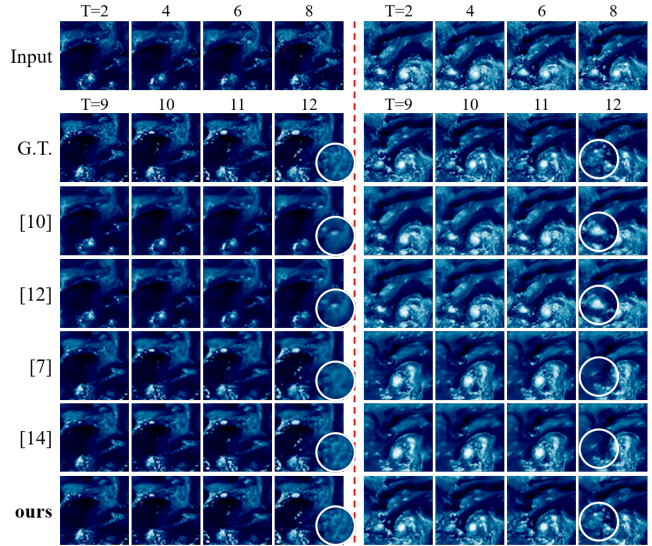

Fig. 4: Qualitative comparisons of previous SOTA models on Typhoon test set at $100,000$ iterations.

Models	PSNR ↑	SSIM ↑	MSE ↓	MAE ↓
ConvLSTM [5]	26.353	0.851	10.43	119.6
PredRNN [10]	27.637	0.887	7.71	107.3
PredRNN++ [12]	28.287	0.891	6.72	114.5
MIM* [7]	26.721	0.893	9.14	132.2
SA-ConvLSTM [14]	28.456	0.898	7.07	94.2
MoDeRNN (ours)	**29.446**	**0.910**	**6.06**	**83.1**

Table 2: Quantitative comparisons of previous SOTA models on Typhoon test set. All models predict the next 4 frames via 8 observed meteorological data.

14]. Frame-wise PSNR, SSIM, MSE, and MAE are adopted to evaluate these models' performance qualitatively and quantitatively, corresponding to Fig.4 and Tab.2.

Tab.2 and Fig.4 demonstrate the proposed method outperforms existing techniques quantitatively and qualitatively. MoDeRNN is the only model that performs well in the detail texture of over timestamps, which enables to preserve and predict the potential trend of meteorological information.

4. CONCLUSION

This paper introduces the novel MoDeRNN for ST-PL, which focuses on tackling the challenging motion trends towards detailed prediction. We propose MoDeRNN to capture fine-grained spatiotemporal latent features to improve the prediction quality in long-term prediction.

In detail, we propose DCB to make latent states well interacted with fine-grained motion details and ensure the prediction results keep consistent clarity. We demonstrate that MoDeRNN achieves satisfactory performance compared to mainstream methods with the lower computational load on 2 representative datasets.

5. REFERENCES

[1] Adam Lerer, Sam Gross, and Rob Fergus, "Learning physical intuition of block towers by example," pp. 430–438, 2016.

[2] Chelsea Finn, Ian J. Goodfellow, and Sergey Levine, "Unsupervised learning for physical interaction through video prediction," in *NeurIPS 2016*, 2016, pp. 64–72.

[3] Jiahao Su, Wonmin Byeon, Jean Kossaifi, et al., "Convolutional tensor-train LSTM for spatio-temporal learning," in *NeurIPS 2020*, 2020.

[4] Sarthak Bhagat, Shagun Uppal, Zhuyun Yin, et al., "Disentangling multiple features in video sequences using gaussian processes in variational autoencoders," in *ECCV 2020*. 2020, pp. 102–117, Springer.

[5] Xingjian Shi, Zhourong Chen, Hao Wang, et al., "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, pp. 802–810, 2015.

[6] Haixu Wu, Zhiyu Yao, Mingsheng Long, et al., "Motionrnn: A flexible model for video prediction with spacetime-varying motions," *CoRR*, vol. abs/2103.02243, 2021.

[7] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, et al., "Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics," in *CVPR 2019*, 2019, pp. 9154–9162.

[8] Yangli-ao Geng, Qingyong Li, Tianyang Lin, et al., "Lightnet: A dual spatiotemporal encoder network model for lightning prediction," in *SIGKDD 2019*. 2019, pp. 2439–2447, ACM.

[9] Xingjian Shi, Zhihan Gao, Leonard Lausen, et al., "Deep learning for precipitation nowcasting: A benchmark and A new model," in *NeurIPS 2017*, 2017, pp. 5617–5627.

[10] Yunbo Wang, Mingsheng Long, Jianmin Wang, et al., "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms," in *NeurIPS 2017*, 2017, pp. 879–888.

[11] Yunbo Wang, Haixu Wu, Jianjin Zhang, et al., "Predrnn: A recurrent neural network for spatiotemporal predictive learning," *CoRR*, vol. abs/2103.09504, 2021.

[12] Yunbo Wang, Zhifeng Gao, Mingsheng Long, et al., "Predrnn++: Towards A resolution of the deep-in-time dilemma in spatiotemporal predictive learning," pp. 5110–5119, 2018.

[13] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, et al., "Eidetic 3d LSTM: A model for video prediction and beyond," in *ICLR 2019*, 2019.

[14] Zhihui Lin, Maomao Li, Zhuobin Zheng, et al., "Self-attention convlstm for spatiotemporal prediction," in *AAAI 2020*, 2020, pp. 11531–11538.

[15] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, , no. 8, pp. 1735–1780, 1997.

[16] Paul J Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, , no. 10, pp. 1550–1560, 1990.

[17] Yann LeCun, Yoshua Bengio, et al., "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[18] Adam Paszke, Sam Gross, Francisco Massa, et al., "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS 2019*, 2019, pp. 8024–8035.

[19] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, et al., "Scheduled sampling for sequence prediction with recurrent neural networks," pp. 1171–1179, 2015.

[20] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[21] Ilya Loshchilov and Frank Hutter, "Fixing weight decay regularization in adam," *CoRR*, 2017.

[22] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov, "Unsupervised learning of video representations using lstms," in *ICML 2015*, 2015, pp. 843–852.

[23] Yuhei Yamamoto, Kazuhito Ichii, Atsushi Higuchi, et al., "Geolocation accuracy assessment of himawari-8/ahi imagery for application to terrestrial monitoring," *Remote. Sens.*, vol. 12, no. 9, pp. 1372, 2020.

[24] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, et al., "Learning to decompose and disentangle representations for video prediction," in *NeurIPS 2018*, 2018, pp. 515–524.

[25] Wei Yu, Yichao Lu, Steve Easterbrook, et al., "Efficient and information-preserving future frame prediction and beyond," in *ICLR 2020*, 2020.

[26] Jérémie Donà, Jean-Yves Franceschi, sylvain lamprier, et al., "Pde-driven spatiotemporal disentanglement," in *ICLR 2021*, 2021.