Structure Preserving Video Prediction

Jingwei Xu Bingbing Ni* Zefan Li Shuo Cheng Xiaokang Yang
Shanghai Institute for Advanced Communication and Data Science
Shanghai Key Laboratory of Digital Media Processing and Transmission
Shanghai Jiao Tong University, Shanghai 200240, China
{xjwxjw,nibingbing,Leezf,xkyang}@sjtu.edu.cn, acccheng94@gmail.com

Abstract

Despite recent emergence of adversarial based methods for video prediction, existing algorithms often produce unsatisfied results in image regions with rich structural information (i.e., object boundary) and detailed motion (i.e., articulated body movement). To this end, we present a structure preserving video prediction framework to explicitly address above issues and enhance video prediction quality. On one hand, our framework contains a two-stream generation architecture which deals with high frequency video content (i.e., detailed object or articulated motion structure) and low frequency video content (i.e., location or moving directions) in two separate streams. On the other hand, we propose a RNN structure for video prediction, which employs temporal-adaptive convolutional kernels to capture time-varying motion patterns as well as tiny objects within a scene. Extensive experiments on diverse scenes, ranging from human motion to semantic layout prediction, demonstrate the effectiveness of the proposed video prediction approach.

1. Introduction

Video prediction is a long-standing task in computer vision research [19, 30, 41, 18]. Boosted by recent emergence of adversarial learning [21], many work [6, 8, 7] attempt to predict future video frames, targeting at higher perceptual quality (i.e., whether the predicted video looks realistic). For example, the work of [5] considers the video prediction task as a min-max game. MCNet [38] directly combines GAN [11] module into a video prediction framework. However, pixel level prediction still remains a challenging task [27], which requires not only to learn the exact static structure of inputs, *e.g.*, object sketch, but also dynamic motion, *e.g.*, articulated movement pattern. Moreover, many of these static and dynamic structural information are fine-grained (i.e., with very detailed texture or subtle motion), which renders video prediction task even more challenging if object structure and motion information need to be preserved in great detail. As shown in Figure 1, at least the following difficulties exist:

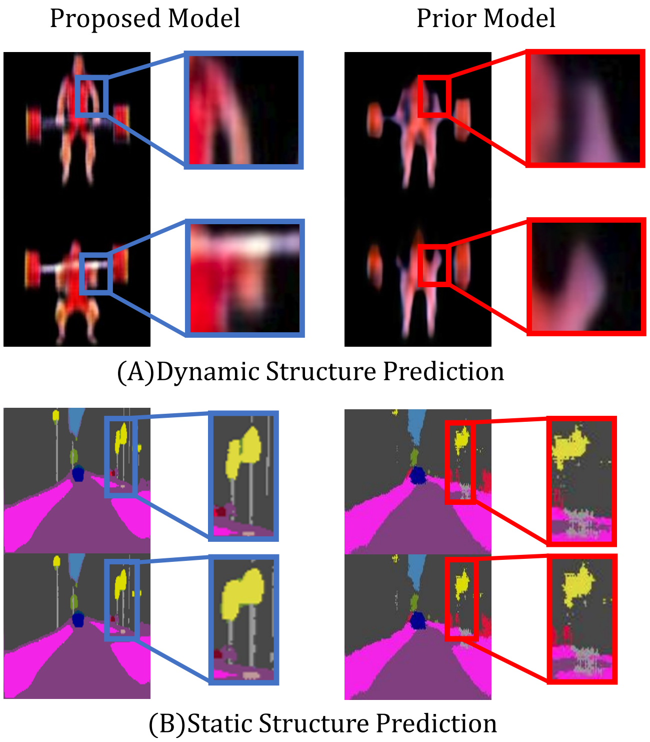
Proposed Model
Prior Model
(A)Dynamic Structure Prediction
(B)Static Structure Prediction

Figure 1. Comparison prediction examples. (A) Dynamic structure: the weightlifting sports consist of complex compound motion, and prior model tends to produce blurry results. (B) Static structure: the traffic sign is very hard to maintain during the prediction because of its slim shape.

- **Static Structure Loss.** This problem mainly arises from predicting these scenes which has a fixed structure, *e.g.*, traffic sign, trees etc. in a city landscape. And the motion of these static structures often result from the motion

[*] Corresponding Author

of camera. Existing methods mostly fail to maintain the original object structures, e.g., detailed boundaries.

- **Dynamic Structure Loss.** Although some recent work [38, 5] are capable of predicting general coarse-grained movements. They generally fails when predicting fine-grained local movements such as articulated motion.

In this paper, we develop an end-to-end framework called structure preserving video prediction net to enhance video prediction. The proposed framework features two components. The first is a **multi-frequency analysis component**. The proposed component contains a high frequency filter, which real-world images passed through and are decomposed into high frequency image part and low frequency image part. Then, multiple predictors are dedicated designed to cope with different frequencies. In particular, a refinement module inspired from the recent image-to-image translation model [16] successfully handles high frequency image part, *i.e*., sketch, to substantially improve the prediction accuracy, i.e., it infers more subtle and realistic object details according to the sketch [42]. The second is a **temporal-adaptive convolution component**. We propose temporal-adaptive convolutional kernels to be embedded in the predictor, which dynamically change the weights based on the accumulated temporal information of video frames (i.e., up to the current processing time step) during prediction. As a result, these kernels could capture time-varying motion patterns, *e.g*., the subtle movements of human limbs involving complex dynamics inside the real-world motion, which are considered very difficult to predict by previous methods [38]. Note that different from Jia *et al*. [17], which only utilizes the current input to generate kernels (i.e., NO temporal memory), our method fully utilizes the temporal variation information to explicitly capture the motion dynamics (i.e., with temporal memory). Both components are integrated in a recurrent neural network based video generation architecture and trained in an end-to-end manner.

We conduct both qualitative and quantitative experiments on diverse datasets, ranging from human motion to semantic layout prediction, including a novel comparison experiment to verify the generalization ability of proposed dynamic prediction scheme, referred as *predicting the past*. These experiments clearly indicate that prediction results of our model could facilitate higher visual quality and more precise prediction even to predict the complex motion patterns as well as detailed object structures (i.e., with tiny objects), which significantly outperform prior arts.

2. Related Work

Video Prediction. Many previous work have been done on video prediction task [40, 36, 25]. Some methods managed to ease the task by introducing some prior knowledge. For example, Denton *et al*. [5] proposed a video prediction model on the basis of the hypothesis that a video sequence could be factored into content and motion. Similarly, Villegas *et al*. [38] proposed a motion-content disentanglement network for pixel-level prediction of future frames in natural video sequences. An action-conditioned video prediction framework developed by Lee *et al*. [30] utilized the action prior knowledge as well as previous appearance information to facilitate future motion prediction. Other methods propose to take extra category information as inputs to facilitate prediction task. For example, multi-task learning was utilized by Liang *et al*. [23] which simultaneously solved the next frame video prediction and optical flow prediction [33] tasks via a dual adversarial training mechanism. Differently, Precup *et al*. [39] proposed a hierarchical approach of pixel-level video prediction, which utilized the human skeleton information to facilitate better prediction quality. Lu *et al*. [24] presents a modular data-driven framework for video prediction based on an end-to-end differentiable network architecture. Recently, Nev *et al*. [29] introduces a new visual understanding task of predicting future semantic segmentations, and proposed a batch model that predicts all future frames at once.

Convolutional LSTM. Recently, the work of [34] proposed a new extension of LSTM called ConvLSTM which had a inherent convolutional scheme within the recurrent architecture. And it had many applications [1, 32, 31] in a large variety of computer vision research area. Marwah *et al*. [26] proposed a video generation framework which utilized the ConvLSTM to encode short-term and long-term spatio-temporal context to generate videos on unseen captions. Stollenga *et al*. [37] proposed a recurrent neural network based on convolutional LSTM that sequentially found objects and their segmentations one at a time. Kalchbrenner *et al*. [19] combined the ConvLSTM into a deep generative model which modelled the factorization of the joint likelihood of inputs in the form of video data. Jia *et al*. [17] introduced a class of dynamic filtering networks, referred as DFN, that applied by dynamically generating filters according to an image.

However, previous video prediction methods tend to produce unsatisfied results when encountered video sequence with rich structural information and complex motion details. And different from above methods, we propose a structure preserving framework which utilizes the high frequency video content and employs temporal-adaptive convolutional kernels to facilitate the video prediction task. Note that our work fully utilizes the motion information between inputs to explicitly capture the spatial-temporal variation of inputs. It is different from the work of [17], which captures the spatial transformation with one frame as the input. Meanwhile, our model use the important mutual information between different channels of the feature map to generate new kernels, but the DFN [17] performs the transformation on each channel of one feature map inde-

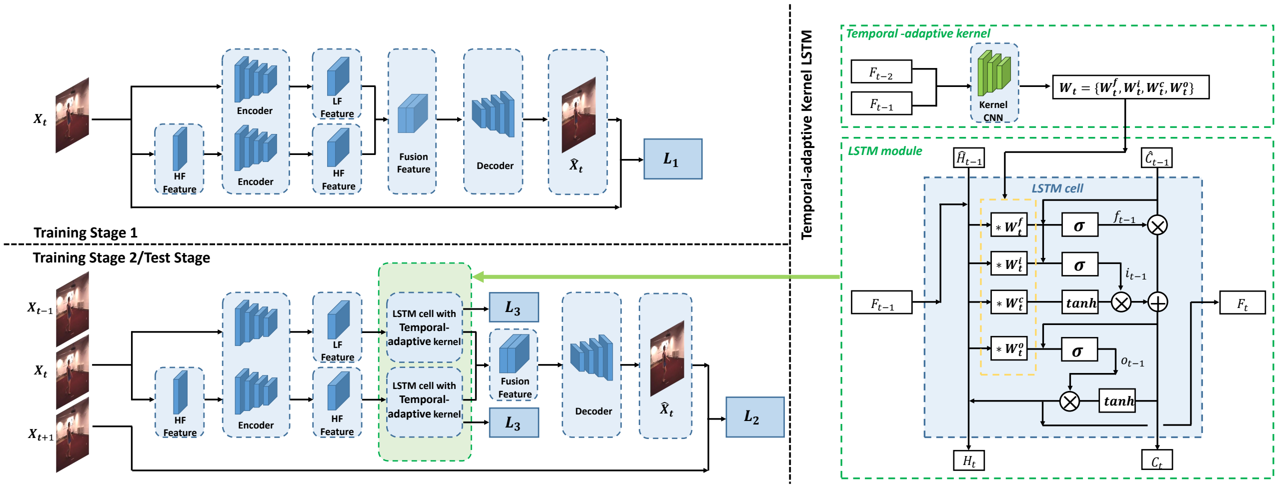

Figure 2. The proposed structure preserving video prediction framework. Left: two-branch framework. At the first training stage we train the encoder and decoder modules, while at the second stage all modules are trained together. Right: temporal-adaptive prediction module.

pendently.

3. Structure Preserving Video Prediction

3.1. Motivation

Previous prediction works are mainly based on the $Encoder - LSTM - Decoder$ architecture [38, 5, 38]. Some of them propose several variant frameworks which employ the prior knowledge over datasets. For example, the MCNet [38] and DrNet [5] disentangle the motion and content parts of video sequence in an unsupervised way for better prediction quality; the work of [39] decomposes video into articulated motion and appearance parts which tackle the task as predicting the low dimension manifold of human motion.

However, these methods are very easy to encounter the following two problems briefly mentioned in Section 1:

Static structure loss. Figure 4(A) demonstrates the predicted results of ConvLSTM [34] on CityScape datasets. We observe that the structure of static objects can not be kept during prediction. For example, the shape of building changes rapidly and the lampposts are missing. This is due to the fact that previous methods ignore the rich structure information contained in the raw-pixel inputs, which could substantially facilitate the static structural prediction. A two-branch video prediction framework is proposed in section 3.2, which contains a multi-frequency analysis module to deal with this problem.

Dynamic structure loss. Figure 4(B) presents the predicted results of ConvLSTM [34] on Human3.6M datasets, which is a human walking sequence. We observe that the predicted results contain severe motion blur compared to the ground truth, *i.e.*, the lower body of human is totally unrecognisable because of the motion blur. This mainly arises from that the moving direction of body parts are different, (*i.e.*, one leg moving forward while the other moving backward). To this end, we propose a temporal adaptive convolution scheme to explicitly solving this problem. Details are given in section 3.3.

3.2. Two-branch video prediction framework

To deal with **static structure loss**, we propose a two-branch video prediction framework. As shown in Figure 2, the whole framework consists of three main modules, *i.e.*, the encoder module, the prediction module and the decoder module. The main contribution in this framework lies in that we use two branches in the encoder and prediction modules to capture different frequency domain information, which boosts the prediction accuracy by a large margin. Note that the high frequency information could be obtained simply passed through a high pass filter. The details of three modules are given as follows.

Encoder module. Let $\mathcal{X} = (\mathcal{X}_1, ..., \mathcal{X}_T)$ denote a video sequence of $T, (= N + M)$ frames in the training set, and our task is to perform M time-steps prediction given N frames as inputs. The two-branch encoders are designed for two different frequency domains. To be specific, the raw pixels are directly passed to the first encoder, denoted as E_L. As for the second branch, we firstly process the raw inputs with a high pass filter, denoted as HF, and then feed the output into another encoder, denoted as E_H. For a T-frame input, the outputs of E_L and E_H are denoted as $\mathcal{F}^L = (\mathcal{F}_1^L, ..., \mathcal{F}_T^L)$ and $\mathcal{F}^H = (\mathcal{F}_1^H, ..., \mathcal{F}_T^H)$ respectively. **In the following we drop the subscript** L **and** H **for brevity**.

Prediction module. In this module, we adopt the seq-to-seq architecture [3] for prediction. We take the N time steps outputs from the encoder modules as the inputs, de-

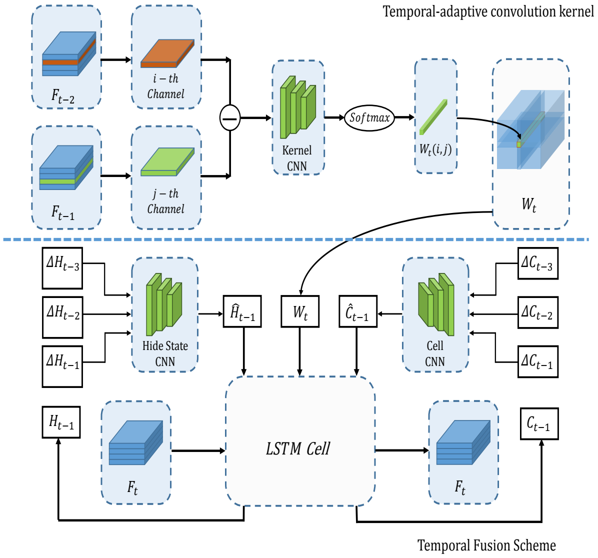

Figure 3. The detailed structure of proposed prediction module. The upper part is proposed temporal-adaptive convolution kernel, **Tem-K**, while the lower part is the temporal fusion scheme,**Fus-4**.

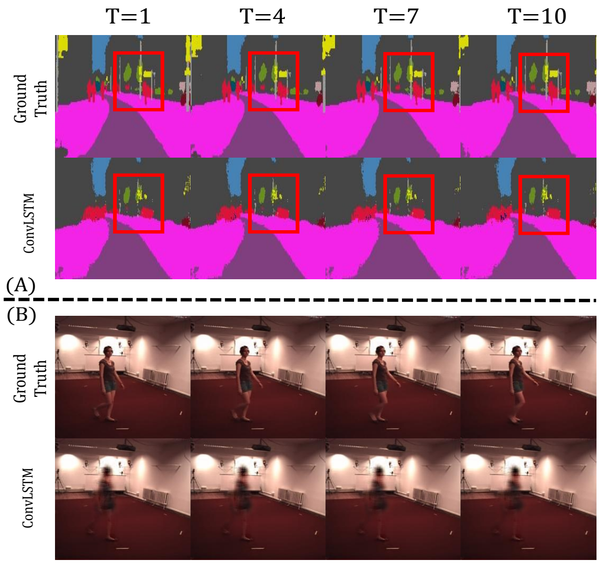

Figure 4. Illustration of structure loss. (A) Static structure loss. (B) Dynamic structure loss. Best view in color.

noted as $(\mathcal{F}_1, ..., \mathcal{F}_N)$. And the first N time steps outputs are denoted as $(\hat{\mathcal{F}}_1, ..., \hat{\mathcal{F}}_N)$. In the following M time steps we sequentially set $\mathcal{F}_{t+1} = \hat{\mathcal{F}}_t, t = N, ..., N + M - 1$ as inputs to produce the final M time steps prediction, denotes as $(\hat{\mathcal{F}}_{N+1}, ..., \hat{\mathcal{F}}_{N+M})$. Meanwhile, inspired from the DenseNet architecture [14], whose dense connection is performed in the channel direction of the CNN, we propose a temporal dense connection scheme. As shown in Figure 3, the hidden state of the last 4 time-steps are first passed through a fusion sub-module, then feed into the next time-step for prediction. By doing so, we aim to purse a more efficient temporal information sharing mechanism to facilitate the video prediction task. More details of this temporal-adaptive convolution module are specified in Section 3.3.

Decoder module. The decoder module takes the outputs of prediction module as inputs. Similar to the recent work [16] on the image to image translation task, our decoder module can also be considered as a refinement module, which utilizes the high frequency information to refine the blurry outputs, *i.e.*, the low frequency prediction, to a more precise version. We train the two-branch encoders and decoder modules together to minimize the regression loss. And formally, let $\hat{\mathcal{X}} = (\hat{\mathcal{X}}_1, ..., \hat{\mathcal{X}}_N, \hat{\mathcal{X}}_{N+1}, ..., \hat{\mathcal{X}}_{N+M})$ denote the outputs of decoder module.

3.3. Temporal adaptive prediction module

To capture the temporal varying motion patterns and deal with the **dynamic structure loss** problem, we propose a novel temporal-adaptive convolution module shown in Figure 3:

This module is developed on the basis of ConvLSTM [34]. We follow the same notation settings except the input feature $\mathcal{F}_t$, the new designed dynamic kernel set $\mathbb{W}_t$ =$\{\mathcal{W}_t^{xi}, \mathcal{W}_t^{hi}, \mathcal{W}_t^{xf}, \mathcal{W}_t^{hf}, \mathcal{W}_t^{xc}, \mathcal{W}_t^{hc}, \mathcal{W}_t^{xo}, \mathcal{W}_t^{ho}\}$, cell input $\hat{\mathcal{C}}_{t-1}$ and hidden state input $\hat{\mathcal{H}}_{t-1}$:

$$\begin{aligned}
i_{t-1} &= \sigma(\mathcal{W}_t^{xi} * \mathcal{F}_{t-1} + \mathcal{W}_t^{hi} * \hat{\mathcal{H}}_{t-1} + \mathcal{W}^{ci} \circ \hat{\mathcal{C}}_{t-1} + b_i), \\
f_{t-1} &= \sigma(\mathcal{W}_t^{xf} * \mathcal{F}_{t-1} + \mathcal{W}_t^{hf} * \hat{\mathcal{H}}_{t-1} + \mathcal{W}^{cf} \circ \hat{\mathcal{C}}_{t-1} + b_f), \\
\mathcal{C}_{t-1} &= f_{t-1} \circ \hat{\mathcal{C}}_{t-1} + i_t \circ \tanh(\mathcal{W}_t^{xc} * \mathcal{F}_{t-1} + \mathcal{W}_t^{hc} * \hat{\mathcal{H}}_{t-1} + b_c), \\
o_{t-1} &= \sigma(\mathcal{W}_t^{xo} * \mathcal{F}_{t-1} + \mathcal{W}_t^{ho} * \hat{\mathcal{H}}_{t-1} + \mathcal{W}^{co} \circ \mathcal{C}_{t-1} + b_o), \\
\mathcal{H}_{t-1} &= o_{t-1} \circ \tanh(\mathcal{C}_{t-1}),
\end{aligned} \tag{1}$$

with $\mathbb{W}_t, \hat{\mathcal{C}}_{t-1}$ and $\hat{\mathcal{H}}_{t-1}$ computed as follow:

$$\begin{aligned}
\mathcal{W}_t &= \phi_{\mathcal{W}_t}(\mathcal{F}_{t-1}; \mathcal{F}_{t-2}), \mathcal{W}_t \in \mathbb{W}_t \\
\hat{\mathcal{H}}_{t-1} &= \phi_{\mathcal{H}}(\Delta\mathcal{H}_{t-1}, \Delta\mathcal{H}_{t-2}, \Delta\mathcal{H}_{t-3}), \\
\hat{\mathcal{C}}_{t-1} &= \phi_{\mathcal{C}}(\Delta\mathcal{C}_{t-1}, \Delta\mathcal{C}_{t-2}, \Delta\mathcal{C}_{t-3}), \\
\Delta\mathcal{H}_{t-i} &= \mathcal{H}_{t-(i+1)} - \mathcal{H}_{t-(i+2)}, i = 1, 2, 3, \\
\Delta\mathcal{C}_{t-i} &= \mathcal{C}_{t-(i+1)} - \mathcal{C}_{t-(i+2)}, i = 1, 2, 3.
\end{aligned} \tag{2}$$

Here $\mathcal{W}t$ denotes one convolution kernel with shape (W_k, H_k, C_I, C_O), where W_k, H_k, C_I, C_O stand for the kernel width, kernel height, input channel and output channel respectively. $\phi_{\mathcal{W}_t}$ is a kernel generation function, denoted as **Tem-K**, designed to fully utilize the temporal variation information while $\phi_{\mathcal{H}}$ and $\phi_{\mathcal{C}}$ are 1-layer CNN, denoted as **Fus-4**, designed to generate hidden state and cell by fusing previous ones.

We use $\mathcal{W}_t(i, j)$ to denote the ith input channel and jth output channel of $\mathcal{W}_t$. $\mathcal{F}_t(i)$ denotes the ith channel of input feature map $\mathcal{F}_t$. Then,

$$\widetilde{\mathcal{W}}_t(i, j) = \widetilde{\phi}_{\mathcal{W}}(\mathcal{F}_t(i) - \mathcal{F}_{t-1}(j)), i, j = 1, ..., C. \tag{3}$$

Here, the kernel generation function $\widetilde{\phi}_{\mathcal{W}}$ is a 3-layer CNN. In contrast to the common convolution operation, which performs channel-wise summation, $\widetilde{\phi}_{\mathcal{W}}$ firstly performs the channel subtraction to obtain the temporal variation information, and then encodes it into the current convolution kernel. Inspired from Jia *et al.* [17], we perform channel-wise softmax [2] along the input channel:

$$\mathcal{W}_t^{\psi}(\cdot, j) = Softmax(\widetilde{W}_t^{\psi}(\cdot, j)), j = 1, ..., C, \quad (4)$$

This increases the sparsity of the generated kernel, which means that majority values within the generated kernel are near 0, while a small portion of them are close to 1. Intuitively, the convolution operation performed by a sparse kernel with binary values of 0 or 1, could perform spatial transformation in a pixel-wise manner. By doing so, we could mimic the complex motion dynamics more precisely. It should be noticed that, the proposed temporal adaptive convolution kernels can be seamlessly integrated into other convolution based recurrent architectures, besides the ConvLSTM [34].

3.4. Implementation details

We give some implementation details in this section.

Loss function design. It is common to get a poor local minima when training a heavy neural network all parts together [28]. The training process for the proposed network is divided into two phases:

At the first phase, we train the encoder together with the decoder modules. We propose a regression loss $\mathcal{L}_1$ to constrain our model in both low and high frequency domain:

$$\mathcal{L}_1 = ||\mathcal{X} - \hat{\mathcal{X}}||_1 + ||HF(\mathcal{X}) - HF(\hat{\mathcal{X}})||_1, \quad (5)$$

where the HF is the high pass filter mentioned in the encoder module of section 3.2. At the second phase, we train the whole framework, keeping the learning rate of the encoder and decoder modules at a relative low values compared to the prediction module. The prediction module is also trained with a regression loss $\mathcal{L}_2$:

$$\begin{aligned}\mathcal{L}_2 = \sum_{i=1}^{N+M-1} & (||\mathcal{X}_{i+1} - \hat{\mathcal{X}}_i||_1 + ||\mathcal{F}_{i+1} - \hat{\mathcal{F}}_i||_1 \\ & + ||HF(\mathcal{X}_{i+1}) - HF(\hat{\mathcal{X}}_i)||_1).\end{aligned} \quad (6)$$

For the dynamic LSTM module, we proposed an additional loss function $\mathcal{L}_3$:

$$\mathcal{L}_3 = \frac{1}{N+M} \sum_{t=1}^{N+M} ||(||\mathcal{F}_t - \hat{\mathcal{F}}_t||_1) - \sigma_{ths}||_1, \quad (7)$$

where the loss term means that we encourage the outputs to be different from the inputs and the σ_{ths} is a predefined threshold with fixed value. Finally, we could train the whole network using the following loss function:

$$\mathcal{L} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \lambda_3 \mathcal{L}_3 + \lambda_4 \sum ||\Theta||_2^2, \quad (8)$$

where the $L2$ regularization term over all the parameters , Θ, is to prevent the model from over-fitting. Details about the network structures and parameters settings are specified as follow:

Two-stream encoder. Both two encoders have three convolution layers and each layer is followed with a leakyReLU [13] layer (the leaky rate is 0.1) as the activation function. All three layers share the same stride of 2 and kernel size 3 × 3, and the output channels are 8, 16, 32 respectively. So the shape of feature maps is (256,256,3)-(128,128,8)-(64,64,16)-(32,32,32). The difference between these two streams lies in that the inputs are first processed by a standard 5 × 5 LoG [12] filter before feed into high frequency encoder.

Temporal-adaptive convLSTM. The kernel size of convLSTM is (9,9,32,32). We use a three-layer convolution network to generate the temporal-adaptive kernel, and each layer is of stride 2 with no activation layer. And we use a single convolution layer to implement the hidden state fusion module, with input shape of (32,32,128) and output shape of (32,32,32). And these two branches share the same architecture.

Decoder. The decoder is implemented with three transpose convolution layers with kernel size 3 × 3 and stride 2. Each layer is followed with a ReLU [10] layer. Note that we concatenate feature maps of these two branches along the channel direction as the inputs. So the shape of feature maps is (32,32,64)-(64,64,16)-(128,128,8)-(256,256,3).

Training procedure. As mentioned in Section 3.2, we train the model at two phases. To be specific, at the first phase we train the encoder and decoder with learning rate of 1e-4 and batch size of 32 for 3 epochs; at the second phase the whole model is trained with batch size of 64 for 10 epochs, while the learning rate of encoder of decoder is reduced to 1e-5, and that of the temporal-adaptive convLSTM is 1e-4. We use the AdamOptimizer [20] during the whole training procedure with $\beta = 0.9$. And the hyper-parameters in Equation 8, $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \sigma_{ths}\}$ is set differently on different datasets. For example, on Human3.6M datasets [15], $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \sigma_{ths}\}$ is $\{1, 5, 10, 0.0001, 0.014\}$, which is fine-tuned on other datasets.

4. Experiments

4.1. Datasets

We evaluate our model on three diverse datasets as follows:

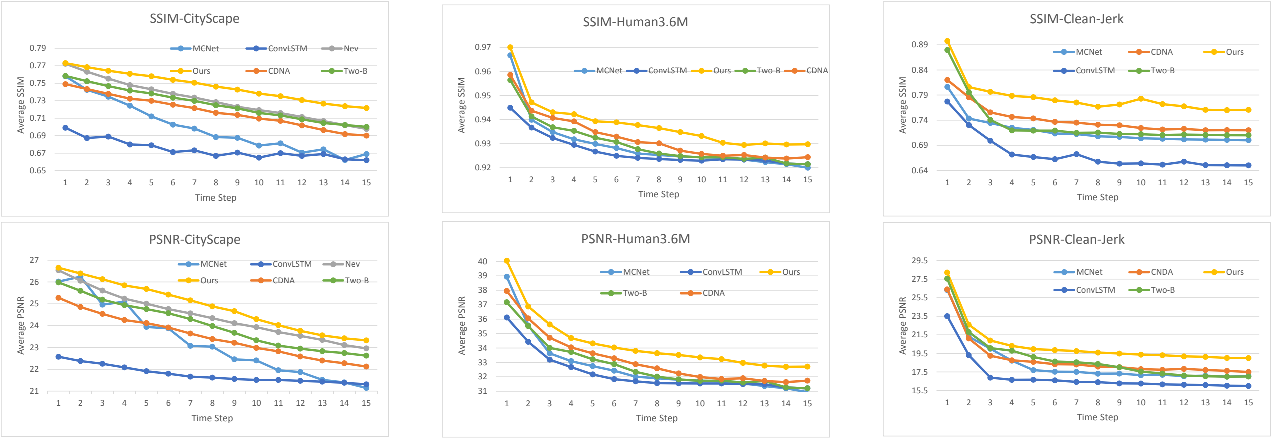

Figure 5. Quantitative comparison of different prediction models on three datasets in term of SSIM and PSNR. Best view in color. There is a noticeable performance drop on Human3.6M Datasets with CDNA [9], *i.e.*, PSNR 42 in original paper and PSNR 38 in our implementation. This mainly results from the difference of target video resolution *i.e.*, 64x64 in original paper and 256x256 in our implementation.

- **UCF-101 Datasets [35]**. This dataset contains videos of athletes practicing 101 different actions. Because of the motion complexity, the future is highly unpredictable. In our experiments, we choose the ”Clean-Jerk” term with 49 video sequences of resolution resized to 256x256. Here we denote the chosen subsets as Clean-Jerk Datasets.

- **Human3.6M Datasets [15]**. This dataset contains a variety of human daily actions with 17 different scenarios. The main difficulties of prediction lie in that human3.6M datasets [15] contain many subtle movements throughout all video sequences, for example random swing of limbs.

- **CityScape Datasets [4]**. This large scale dataset contains 2,975/500 train/val video sequences with 19 semantic classes. We follows the Nev *et al*. [29] to obtain the semantic layouts. **Note that on this dataset our task is to predict the semantic layouts given the previous ones**. And all video sequences are resized to 128x256.

4.2. Baselines and Evaluation Setup

To demonstrate the effectiveness of our proposed model, we compare our model with three strong methods, which are MCNet [38], CDNA [9] and Nev *et al*. [29]. To be specific, MCNet [38] achieves state-of-the-art performance on KTH datasets [22], and CDNA [9] performs best on Human3.6M datasets [15], while the Nev *et al*. [29] is the first work on predicting the semantic layouts of CityScape datasets [4]. To evaluate the **dynamic structure loss**, we mainly compare the prediction results with MCNet [38] as well as CDNA [9] on Human3.6M [15] and Clean-Jerk datasets [35]. But we also demonstrate the results of these two models on CityScape datasets [4] for further evaluation. As for the **static structure loss**, we compare with Nev *et al*. [29] on CityScape datasets [4], considering that it is a dedicated designed model for this dataset. To ensure fair comparison, all models are trained with the configuration reported in their papers.

During evaluation, we perform 10 time steps forward prediction given the previous 10 frames as inputs. We demonstrate the quantitative evaluation in Section 4.3, which includes PSNR and SSIM, commonly used to evaluate the general performance in previous works [38, 9, 29]. Meanwhile we take two baselines for comparison: the first one is the ConvLSTM [34], and the second one is proposed two-branch framework, denoted as **Two-B**. While the qualitative evaluation includes two aspects: one is to verify the static structure preservation ability (mainly compared with Nev *et al*. [29]); another is to examine the dynamic prediction ability (mainly compared with MCNet [38] and CDNA [9]). Details are given in Section 4.4.

4.3. Quantitative Evaluation

In the quantitative experiments, our goal is to verify whether our model infers more reasonable future under these evaluation metrics.

Figure 5 illustrates the quantitative results of our models compared to prior methods. Note that all models produce ten time steps prediction during training, but we demonstrate the fifteen prediction results to verify the generalization ability of these models. From Figure 5 we have several observations:

First, the MCNet [38] as well as CDNA [9] achieve promising prediction results on the articulated motion prediction, *i.e.*, Human3.6M datasets [15], but do not perform well on CityScape datasets [4]. This mainly results from that these two methods does not take complex structure information into consideration during prediction.

Second, the Nev *et al*. [29] generally performs better than than MCNet [38] and CDNA [9] on CityScape datasets [4], which splits the inputs into multiple spatial scales, *i.e.*, formulating a coarse-to-fine structure, to facilitate structure prediction.

Third, our model outperforms other three models by a

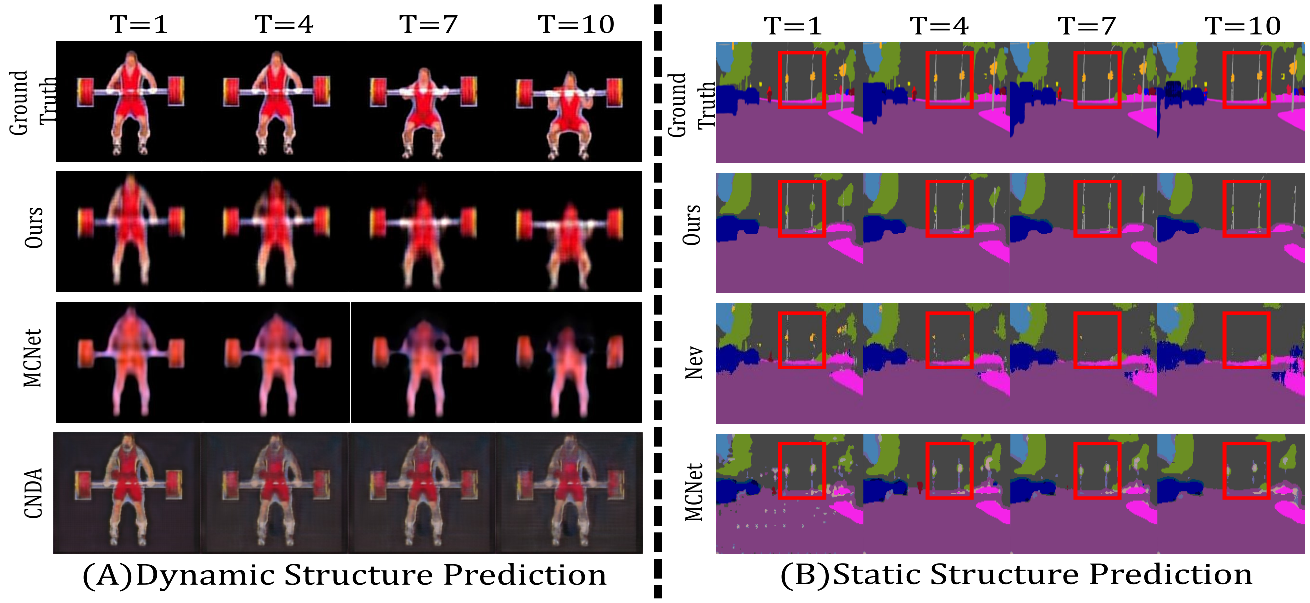

Figure 6. Prediction results of different models. (A) Dynamic structure prediction. (B) Static structure prediction. Best view in color.

large margin, thanks to the usage of two kinds of information, *i.e.*, the dynamic motion as well as static structure to purse better prediction quality. And the performance of proposed model degrades more gracefully throughout the ten frames prediction compared to prior work, *e.g.*, MCNet [38] on CityScape datasets [4], which clearly demonstrates the robustness of our model.

4.4. Qualitative Evaluation

In the qualitative evaluation, we demonstrate the prediction results on different models to address these two issues mentioned in Section 3.1, *i.e.* dynamic structure loss and static structure loss.

The Figure 6 (A) presents the prediction results at 4 time steps on Clean-Jerk datasets [35]. From the top to bottom we sequentially show the ground truth, the results of our proposed model, MCNet [38] and CDNA [9]. It should be noticed that the compound motion of athlete and bell actually forms a complex temporal dynamic structure. And we observe that MCNet [38] makes it to capture the general movements of the athlete (the third row), *i.e.*, the going down motion, but the predicted frames are blurry and the structure of the athlete is incomplete, *i.e.*, the upper body is almost missed out at time step ten. The last row shows the prediction results of CDNA [9], whose the visual quality of is relative higher than that of MCNet [38], but this model do not capture the dynamic structure of the both the athlete and bell, whose prediction is nearly stuck at all time steps. Different from these two models, the temporal-adaptive convolution module of our model successfully captures the dynamic structure of both two subjects and precisely predicts the moving direction of them. Meanwhile benefiting from the multi-frequency analysis module which utilizes different frequencies information of the inputs, the predicted frames are visually satisfying, *i.e.*, detailed structures are preserved.

Model	CityScape/Human3.6M/Clean-Jerk	
	PSNR	SSIM
ConvLSTM	22.8/36.2/23.4	0.70/0.94/0.78
Two-B	25.2/37.2/25.3	0.74/0.96/0.85
Two-B+Fus-4	25.7/37.5/25.7	0.76/0.96/0.85
Two-B+Fus-4+Tem-K	**26.6/39.7/27.5**	**0.77/0.97/0.89**

Table 1. Ablation study of the proposed model.

We report the prediction results of CityScape datasets [4] The Figure 6 (B). And we compare our model with Nev *et al.* [29] as well as MCNet [38]. We observe that the slim traffic sign is very difficult for Nev *et al.* [29] to predict. As shown in the third row at Figure 6 (B), the traffic sign can be only predicted by Nev *et al.* [29] at the first time step. The prediction results of MCNet [38] are slightly better, *i.e.*, the lower part of the traffic sign is maintained, but still incomplete compared to ground truth. In contrast, our proposed model successfully predicts the traffic sign at all time steps, which clearly proves that the high frequency information is a crucial reference for the static structure prediction. We strongly suggest the readers to refer the supplementary material for more examples.

4.5. Ablation Study

Figure 7 demonstrates results of an ablation study of our proposed model, assessing the influence of all components we use. Here we use two baselines for comparison. The first one is the ConvLSTM [34], and the second one is the

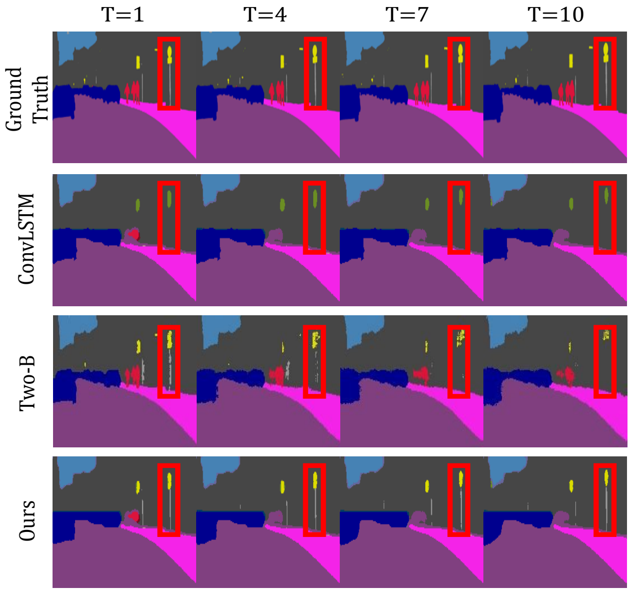

Figure 7. Ablation study on CityScape datasets [4]. Best view in color.

proposed two-branch framework, denoted as **Two-B**. From Figure 7 we observe that the ConvLSTM model fails to capture the slim traffic sign during prediction. In contrast, the two-branch framework is able to capture this kind of static structure in the short-term prediction (the third row in Figure 7), *i.e.*, predicting the next frame, but fails to maintain it in the long-term prediction, *i.e.*, prediction the following ten frames. On the basis of the two-branch framework, the temporal-adaptive kernel makes it to predict long-term variation of the traffic sign (the fourth row in Figure 7). This indicates that the two-branch module is able to cope with the temporal-adaptive module to facilitate higher prediction accuracy in dynamically changing objects.

As illustrated in Table 1, the quantitative results on three datasets are presented for comparison. Note that these results are averaged on the whole test set of the first frame prediction. Both the two-branch module (Two-B) and the temporal-adaptive module (Tem-K) improve the prediction accuracy by a large margin. The hidden state fusion scheme (Fus-4) contributes the enhancement on the CityScape datasets [4], while it does not lead to a significant improvement on the other two.

4.6. Predicting the Past

To verify the generalization ability the temporal-adaptive convolution module, we conduct a novel comparison experiment, referred as predicting the past. To be specific, the models are trained with temporal sequential order, but are tested with temporal reverse order, which increases the prediction difficulty because the temporal motion pattern is never met by the models.

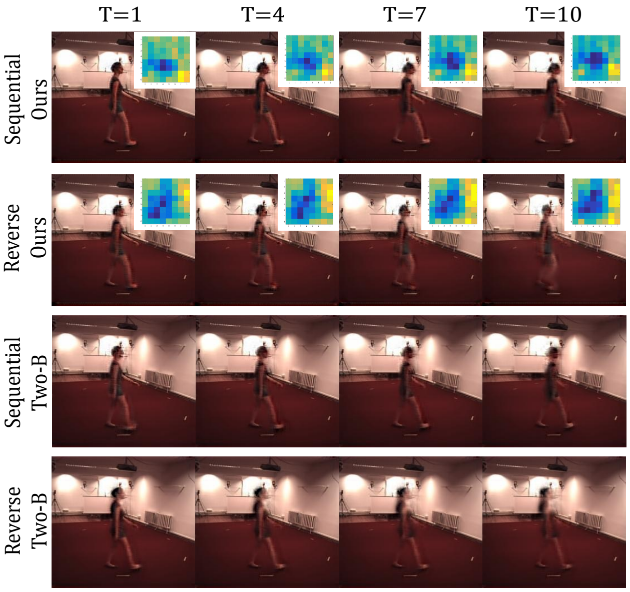

Figure 8. Examples of predicting the past experiment. Best view in color.

Figure 8 demonstrates two temporal orders (*i.e.*, sequential and reverse) prediction results. The top two rows correspond to these results of proposed structure preserving model, and the up right conner of each result represents the current generated 9×9 kernel. We select the $\mathcal{W}^{xi}(12, 12)$ for visualization. The bottom two rows present these of the two-branch architecture (*i.e.*, without the temporal-adaptive module). Here we have several observations: (1) the two-branch architecture fails to give reasonable prediction on the reverse order (the fourth row), *i.e.*, the lower body is totally stuck; (2) the generated kernel changes with the variation of current inputs on both two orders, and brighter color indicates higher value, whose sparsity verifies the hypothesis in section 3.3; (3) the proposed structure preserving framework produces sensible motion even on the reverse order (the second row), which clearly shows the generalization ability of the temporal-adaptive convolution module.

5. Conclusion

In this paper, we present a structure preserving video prediction framework to explicitly address the static and dynamic structure loss issues. Extensive experiments demonstrate the effectiveness of the proposed video prediction approach.

6. Acknowledgement

This work was supported by National Science Foundation of China (U1611461, 61502301,61671298,61521062). The work was partially supported by State Key Research and Development Program (2016YFB1001003), China's Thousand Youth Talents Plan, STCSM 17511105401 and 18DZ2270700.

References

[1] N. Ballas, L. Yao, C. Pal, and A. C. Courville. Delving deeper into convolutional networks for learning video representations. *CoRR*, abs/1511.06432, 2015. 2

[2] C. M. Bishop. Pattern recognition and machine learning. 5

[3] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. 3

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6, 7, 8

[5] E. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. *CoRR*, abs/1705.10915, 2017. 1, 2, 3

[6] E. L. Denton, S. Chintala, a. szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1486–1494. Curran Associates, Inc., 2015. 1

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 1

[8] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[9] C. Finn, I. J. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 64–72, 2016. 6, 7

[10] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 315–323, 2011. 5

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 1

[12] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1992. 5

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034, 2015. 5

[14] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. 4

[15] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 5, 6

[16] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 2, 4

[17] X. Jia, B. D. Brabandere, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 667–675, 2016. 2, 5

[18] X. Jin, H. Xiao, X. Shen, J. Yang, Z. Lin, Y. Chen, Z. Jie, J. Feng, and S. Yan. Predicting Scene Parsing and Motion Dynamics in the Future. *ArXiv e-prints*, Nov. 2017. 1

[19] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1771–1779, 2017. 1, 2

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[21] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. 1

[22] I. Laptev and T. Lindeberg. Space-time interest points. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 432–439, 2003. 6

[23] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual motion GAN for future-flow embedded video prediction. *CoRR*, abs/1708.00284, 2017. 2

[24] C. Lu, M. Hirsch, and B. Schölkopf. Flexible spatio-temporal networks for video prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017. 2

[25] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[26] T. Marwah, G. Mittal, and V. N. Balasubramanian. Attentive semantic video generation using captions. *CoRR*, abs/1708.05980, 2017. 2

[27] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015. 1

[28] D. J. Montana and L. Davis. Training feedforward neural networks using genetic algorithms. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989*, pages 762–767, 1989. 5

[29] N. Neverova, P. Luc, C. Couprie, J. J. Verbeek, and Y. LeCun. Predicting deeper into the future of semantic segmentation. *CoRR*, abs/1703.07684, 2017. 2, 6, 7

[30] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. P. Singh. Action-conditional video prediction using deep networks in atari

games. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2863–2871, 2015. 1, 2

[31] P. Ondruska and I. Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. *CoRR*, abs/1602.00991, 2016. 2

[32] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *CoRR*, abs/1511.06309, 2015. 2

[33] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised deep learning for optical flow estimation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1495–1501, 2017. 2

[34] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015. 2, 3, 4, 5, 6, 7

[35] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 6, 7

[36] R. Stewart and S. Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 2576–2582, 2017. 2

[37] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber. Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2998–3006. Curran Associates, Inc., 2015. 2

[38] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *CoRR*, abs/1706.08033, 2017. 1, 2, 3, 6, 7

[39] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3560–3569, 2017. 2, 3

[40] N. Watters, A. Tacchetti, T. Weber, R. Pascanu, P. Battaglia, and D. Zoran. Visual interaction networks. *CoRR*, abs/1706.01433, 2017. 2

[41] Y. Yoo, K. Yun, S. Yun, J. Hong, H. Jeong, and J. Young Choi. Visual path prediction in complex scenes with crowded moving objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[42] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. 2