Learning to Generate Long-term Future via Hierarchical Prediction

Ruben Villegas [1 *] **Jimei Yang** [2] **Yuliang Zou** [1] **Sungryull Sohn** [1] **Xunyu Lin** [3] **Honglak Lee** [1 4]

Abstract

We propose a hierarchical approach for making long-term predictions of future frames. To avoid inherent compounding errors in recursive pixel-level prediction, we propose to first estimate high-level structure in the input frames, then predict how that structure evolves in the future, and finally by observing a single frame from the past and the predicted high-level structure, we construct the future frames without having to observe any of the pixel-level predictions. Long-term video prediction is difficult to perform by recurrently observing the predicted frames because the small errors in pixel space exponentially amplify as predictions are made deeper into the future. Our approach prevents pixel-level error propagation from happening by removing the need to observe the predicted frames. Our model is built with a combination of LSTM and analogy-based encoder-decoder convolutional neural networks, which independently predict the video structure and generate the future frames, respectively. In experiments, our model is evaluated on the Human3.6M and Penn Action datasets on the task of long-term pixel-level video prediction of humans performing actions and demonstrate significantly better results than the state-of-the-art.

1. Introduction

Learning to predict the future has emerged as an important research problem in machine learning and artificial intelligence. Given the great progress in recognition (e.g., (Krizhevsky et al., 2012; Szegedy et al., 2015)), prediction becomes an essential module for intelligent agents to plan actions or to make decisions in real-world application scenarios (Jayaraman & Grauman, 2015; 2016; Finn et al., 2016). For example, robots can quickly learn manipulation skills when predicting the consequences of physical interactions. Also, an autonomous car can brake or slow down when predicting a person walking across the driving lane. In this paper, we investigate long-term future frame prediction that provides full descriptions of the visual world.

Recent recursive approaches to pixel-wise video prediction highly depend on observing the generated frames in the past to make predictions further into the future (Oh et al., 2015; Mathieu et al., 2016; Goroshin et al., 2015; Srivastava et al., 2015; Ranzato et al., 2014; Finn et al., 2016; Villegas et al., 2017; Lotter et al., 2017). In order to make reasonable long-term frame predictions in natural videos, these approaches need to be highly robust to pixel-level noise. However, the noise amplifies quickly through time until it overwhelms the signal. It is common that the first few prediction steps are of decent quality, but then the prediction degrades dramatically until all the video context is lost. Other existing works focus on predicting high-level semantics, such as motion trajectories or action labels (Walker et al., 2014; Yuen & Torralba, 2010; Lee, 2015), driven by immediate applications (e.g., video surveillance). We note that such high-level representations are the major factors for explaining the pixel variations into the future. In this work we assume that the high-dimensional video data is generated from low-dimensional high-level structures, which we hypothesize will be critical for making long-term visual predictions. Our main contribution is the hierarchical approach for video prediction that involves generative modeling of video using high-level structures. Concretely, our algorithm first estimates high-level structures of observed frames, and then predicts their future states, and finally generates future frames conditioned on predicted high-level structures.

The prediction of future structure is performed by an LSTM that observes a sequence of structures estimated by a CNN, encodes the observed dynamics, and predicts the future sequence of such structures. We note that Fragkiadaki et al. (2015) developed an LSTM architecture that can straightforwardly be adapted to our method. However, our main contribution is the hierarchical approach for video prediction, so we choose a simpler LSTM architecture to convey our idea. Our approach then observes a single frame from the past and predicts the entire future described by the predicted structure sequence using an analogy-making

[*] Work completed while at Google Brain. [1]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. [2]Adobe Research, San Jose, CA. [3]Beihang University, Beijing, China. [4]Google Brain, Mountain View, CA. Correspondence to: Ruben Villegas <rubville@umich.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

network (Reed et al., 2015). In particular, we propose an image generator that learns a shared embedding between image and high-level structure information which allows us convert an input image into a future image guided by the structure difference between the input image and the future image. We evaluate the proposed model on challenging real-world human action video datasets. We use 2D human poses as our high-level structures similar to Reed et al. (2016a). Thus, our LSTM network models the dynamics of human poses while our analogy-based image generator network learns a joint image-pose embedding that allows the pose difference between an observed frame and a predicted frame to be transferred to image domain for future frame generation. As a result, this pose-conditioned generation strategy prevents our network from propagating prediction errors through time, which in turn leads to very high quality future frame generation for long periods of time. Overall, the promising results of our approach suggest that it can be greatly beneficial to incorporate proper high-level structures into the generative process.

The rest of the paper is organized as follows: A review of the related work is presented in Section 2. The overview of the proposed algorithm is presented in Section 3. The network configurations and their training algorithms are described in Section 4 and Section 5, respectively. We present the experimental details and results in Section 6, and conclude the paper with discussions of future work in Section 7.

2. Related Work

Early work on future frame prediction focused on small patches containing simple predictable motions (Sutskever et al., 2009; Michalski et al., 2014; Mittelman et al., 2014) and motions in real videos (Ranzato et al., 2014; Srivastava et al., 2015). High resolution videos contain far more complicated motion which cannot be modeled in a patch-wise manner due to the well known aperture problem. The aperture problem causes blockiness in predictions as we move forward in time. Ranzato et al. (2014) tried to solve blockiness by averaging over spatial displacements after predicting patches; however, this approach does not work for long-term predictions.

Recent approaches in video prediction have moved from predicting patches to full frame prediction. Oh et al. (2015) proposed a network architecture for action conditioned video prediction in Atari games. Mathieu et al. (2016) proposed an adversarial loss for video prediction and a multi-scale network architecture that results in high quality prediction for a few timesteps in natural video; however, the frame prediction quality degrades quickly. Finn et al. (2016) proposed a network architecture to directly transform pixels from a current frame into the next frame by predicting a distribution over pixel motion from previous frames. Xue et al. (2016) proposed a probabilistic model for predicting possible motions of a single input frame by training a motion encoder in a variational autoencoder approach. Vondrick et al. (2016) built a model that generates realistic looking video by separating background and foreground motion. Villegas et al. (2017) improved the convolutional encoder/decoder architecture by separating motion and content features. Lotter et al. (2017) built an architecture inspired by the predictive coding concept in neuroscience literature that predicts realistic looking frames.

All the previously mentioned approaches attempt to perform video generation in a pixel-to-pixel process. We aim to perform the prediction of future frames in video by taking a hierarchical approach of first predicting high-level structure and then using the high-level structure to predict the future in the video from a single frame input.

To the best of our knowledge, this is the first hierarchical approach to pixel-level video prediction. Our hierarchical architecture makes it possible to generate good quality long-term predictions that outperform current approaches. The main success from our algorithm comes from the novel idea of first making high-level structure predictions which allows us to observe a single image and generate the future video by visual-structure analogy. Our image generator learns a shared embedding between image and structure inputs that allows us to transform high-level image features into a future image driven by the predicted structure sequence.

3. Overview

This paper tackles the task of long-term video prediction in a hierarchical perspective. Given the input high-level structure sequence $\mathbf{p}_{1:t}$ and frame $\mathbf{x}_t$, our algorithm is asked to predict the future structure sequence $\mathbf{p}_{t+1:t+T}$ and subsequently generate frames $\mathbf{x}_{t+1:t+T}$. The problem with video frame prediction originates from modeling pixels directly in a sequence-to-sequence manner and attempting to generate frames in a recurrent fashion. Current state-of-the-art approaches recurrently observe the predicted frames, which causes rapidly increasing error accumulation through time. Our objective is to avoid having to observe generated future frames at all during the full video prediction procedure.

Figure 1 illustrates our hierarchical approach. Our full pipeline consists of 1) performing high-level structure estimation from the input sequence, 2) predicting a sequence of future high-level structures, and 3) generating future images from the predicted structures by visual-structure analogy-making given an observed image and the predicted structures. We explore our idea by performing pixel-level video prediction of human actions while treating human pose as the high-level structure. Hourglass network (Newell et al., 2016) is used for pose estimation on input images. Subsequently, a sequence-to-sequence LSTM-recurrent network is trained to read the outputs of hourglass network and to

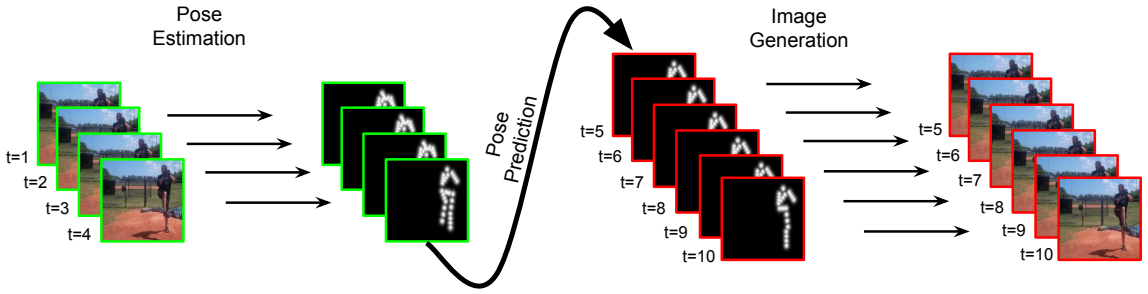

Figure 1. Overall hierarchical approach to pixel-level video prediction. Our algorithm first observes frames from the past and estimate the high-level structure, in this case human pose xy-coordinates, in each frame. The estimated structure is then used to predict the future structures in a sequence to sequence manner. Finally, our algorithm takes the last observed frame, its estimated structure, and the predicted structure sequence, in this case represented as heatmaps, and generates the future frames. Green denotes input to our network and red denotes output from our network.

predict the future pose sequence. Finally, we generate the future frames by analogy making using pose relationship in feature space to transform the last observed frame.

The proposed algorithm makes it possible to decompose the task of video frame prediction to sub-tasks of future high-level structure prediction and structure-conditioned frame generation. By doing so, we remove the recursive dependency of generated frames which have caused the compound errors of pixel-level prediction in previous methods, which allows us to perform very long-term video prediction.

4. Architecture

This section describes the architecture of the proposed algorithm using human pose as a high-level structure. Our full network is composed of two modules: an encoder-decoder LSTM that observes and outputs xy-coordinates, and an image generator that performs visual analogy-based on high-level structure heatmaps constructed from the xy-coordinates output from LSTM.

4.1. Future Prediction of High-Level Structures

Figure 2 illustrates our pose predictor. Our network first encodes the observed structure dynamics by

$$[\mathbf{h}_t, \mathbf{c}_t] = \text{LSTM}\left(\mathbf{p}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right), \tag{1}$$

where $\mathbf{h}_t \in \mathbb{R}^H$ represents the observed dynamics up to time t, $\mathbf{c}_t \in \mathbb{R}^H$ is the *memory cell* that retains information from the history of pose inputs, $\mathbf{p}_t \in \mathbb{R}^{2L}$ is the pose at time t (i.e., 2D coordinate positions of L joints). In order to make a reasonable prediction of the future pose, LSTM has to first observe a few pose inputs to identify the type of motion occurring in the pose sequence and how it is changing over time. LSTM also has to be able to remove noise present in the input pose, which can come from annotation error if using the dataset-provided pose annotation or pose estimation error if using a pose estimation algorithm. After a few pose inputs have been observed, LSTM generates the future pose by

$$\hat{\mathbf{p}}_t = f\left(\mathbf{w}^\top \mathbf{h}_t\right), \tag{2}$$

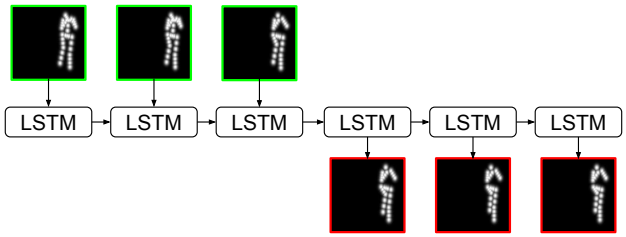

Figure 2. Illustration of our pose predictor. LSTM observes k consecutive human pose inputs and predicts the pose for the next T timesteps. Note that the human heatmaps are used for illustration purposes, but our network observes and outputs xy-coordinates.

where $\mathbf{w}$ is a projection matrix, f is a function on the projection (i.e. tanh or identity), and $\hat{\mathbf{p}}_t \in \mathbb{R}^{2L}$ is the predicted pose. In the subsequent predictions, our LSTM does not observe the previously generated pose. Not observing generated pose in LSTM prevents errors in the pose prediction from being propagated into the future, and it also encourages the LSTM internal representation to contain robust high-level features that allow it to generate the future sequence from only the original observation. As a result, the representation obtained in the pose input encoding phase must obtain all the necessary information for generating the correct action sequence in the decoding phase. After we have set the human pose sequence for the future frames, we proceed to generate the pixel-level visual future.

4.2. Image Generation by Visual-Structure Analogy

To synthesize the future frame given its pose structure, we make a visual-structure analogy inspired by Reed et al. (2015) following $\mathbf{p}_t : \mathbf{p}_{t+n} :: \mathbf{x}_t : \mathbf{x}_{t+n}$, read as "$\mathbf{p}_t$ is to $\mathbf{p}_{t+n}$ as $\mathbf{x}_t$ is to $\mathbf{x}_{t+n}$" as illustrated in Figure 3. Intuitively, the future frame $\mathbf{x}_{t+n}$ can be generated by transferring the structure transformation from $\mathbf{p}_t$ to $\mathbf{p}_{t+n}$ to the observed frame $\mathbf{x}_t$. Our image generator instantiates this idea using a pose encoder f_{pose}, an image encoder f_{img} and an image decoder f_{dec}. Specifically, f_{pose} is a convolutional encoder that specializes on identifying key pose features from the

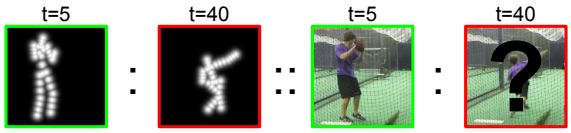

Figure 3. Generating image frames by making analogies between high-level structures and image pixels.

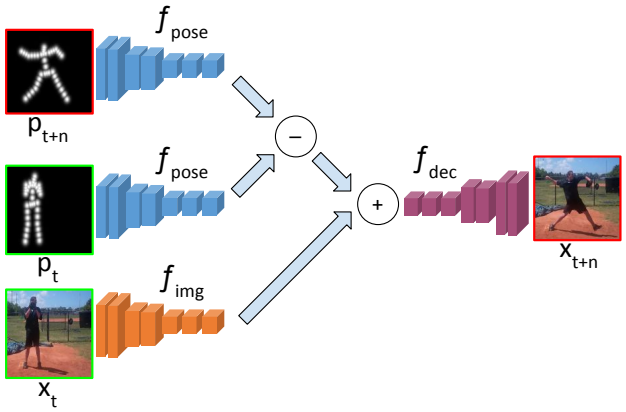

Figure 4. Illustration of our image generator. Our image generator observes an input image, its corresponding human pose, and the human pose of the future image. Through analogy making, our network generates the next frame.

pose input that reflects high-level human structure.[1] f_{img} is also a convolutional encoder that acts on an image input by mapping the observed appearance into a feature space where the pose feature transformations can be easily imposed to synthesize the future frame using the convolutional decoder f_{dec}. The visual-structure analogy is then performed by

$$\hat{\mathbf{x}}_{t+n} = f_{\text{dec}}\left(f_{\text{pose}}\left(g\left(\hat{\mathbf{p}}_{t+n}\right)\right) - f_{\text{pose}}\left(g\left(\mathbf{p}_t\right)\right) + f_{\text{img}}\left(\mathbf{x}_t\right)\right), \tag{3}$$

where $\hat{\mathbf{x}}_{t+n}$ and $\hat{\mathbf{p}}_{t+n}$ are the generated image and corresponding predicted pose at time $t+n$, $\mathbf{x}_t$ and $\mathbf{p}_t$ are the input image and corresponding estimated pose at time t, and $g(.)$ is a function that maps the output xy-coordinates from LSTM into depth-concatenated L heatmaps.[2] Intuitively, f_{pose} infers features whose "substractive" relationship is the same subtractive relationship between $\mathbf{x}_{t+n}$ and $\mathbf{x}_t$ in the feature space computed by f_{img}, i.e., $f_{\text{pose}}(g(\hat{\mathbf{p}}_{t+n})) - f_{\text{pose}}(g(\hat{\mathbf{p}}_t)) \approx f_{\text{img}}(\mathbf{x}_{t+n}) - f_{\text{img}}(\mathbf{x}_t)$. The network diagram is illustrated in in Figure 4. The relationship discovered by our network allows for highly non-linear transformations between images to be inferred by a simple addition/subtraction in feature space.

5. Training

In this section, we first summarize the multi-step video prediction algorithm using our networks and then describe the training strategies of the high-level structure LSTM and of the visual-structure analogy network. We train our high-level structure LSTM independent from the visual-structure analogy network, but both are combined during test time to perform video prediction.

[1]Each input pose to our image generator is converted to concatenated heatmaps of each landmark before computing features.

[2]We independently construct the heatmap with a Gaussian function around the xy-coordinates of each landmark.

Algorithm 1 Video Prediction Procedure

input: $\mathbf{x}_{1:k}$
output: $\hat{\mathbf{x}}_{k+1:k+T}$
for t=1 to k **do**
  $\mathbf{p}_t \leftarrow \text{Hourglass}(\mathbf{x}_t)$
  $[\mathbf{h}_t, \mathbf{c}_t] \leftarrow \text{LSTM}(\mathbf{p}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1})$
end for
for t=$k+1$ to $k+T$ **do**
  $[\mathbf{h}_t, \mathbf{c}_t] \leftarrow \text{LSTM}(\mathbf{h}_{t-1}, \mathbf{c}_{t-1})$
  $\hat{\mathbf{p}}_t \leftarrow f\left(\mathbf{w}^\top \mathbf{h}_t\right)$
  $\hat{\mathbf{x}}_t \leftarrow f_{\text{dec}}\left(f_{\text{pose}}\left(g\left(\hat{\mathbf{p}}_t\right)\right) - f_{\text{pose}}\left(g\left(\mathbf{p}_k\right)\right) + f_{\text{img}}\left(\mathbf{x}_k\right)\right)$
end for

5.1. Multi-Step Prediction

Our algorithm multi-step video prediction procedure is described in Algorithm 1. Given input video frames, we use the Hourglass network (Newell et al., 2016) to estimate the human poses $\mathbf{p}_{1:k}$. High-level structure LSTM then observes $\mathbf{p}_{1:k}$, and proceeds to generate a pose sequence $\hat{\mathbf{p}}_{k+1:k+T}$ where T is the desired number of time steps to predict. Next, our visual-structure analogy network takes $\mathbf{x}_k$, $\mathbf{p}_k$, and $\hat{\mathbf{p}}_{k+1:k+T}$ and proceeds to generate future frames $\hat{\mathbf{x}}_{k+1:k+T}$ one by one. Note that the future frame prediction is performed by observing pixel information from only $\mathbf{x}_k$, that is, we never observe any of the predicted frames.

5.2. High-Level Structure LSTM Training

We employ a sequence-to-sequence approach to predict the future structures (i.e. future human pose). Our LSTM is *unrolled* for k timesteps to allow it to observe k pose inputs before making any prediction. Then we minimize the prediction loss defined by

$$\mathcal{L}_{\text{pose}} = \frac{1}{TL}\sum_{t=1}^{T}\sum_{l=1}^{L} \mathbb{1}_{\{m^l_{k+t}=1\}} \|\hat{\mathbf{p}}^l_{k+t} - \mathbf{p}^l_{k+t}\|_2^2, \tag{4}$$

where $\hat{\mathbf{p}}^l_{k+t}$ and $\mathbf{p}^l_{k+t}$ are the predicted and ground-truth pose l-th landmark, respectively, $\mathbb{1}_{\{.\}}$ is the indicator function, and m^l_{k+t} tells us whether a landmark is visible or not (i.e. not present in the ground-truth). Intuitively, the indicator function allows our LSTM to make a guess of the non-visible landmarks even when not present at training. Even in the absence of a few landmarks during training, LSTM is able to internally understand the human structure and observed motion. Our training strategy allows LSTM to make a reasonable guess of the landmarks not present in the training data by using the landmarks available as context.

5.3. Visual-Structure Analogy Training

Training our network to transform an input image into a target image that is too close in image space can lead to sub-optimal parameters being learned due to the simplicity of such task that requires only changing a few pixels. Because of this, we train our network to perform random *jumps in time* within a video clip. Specifically, we let our network observe a frame $\mathbf{x}_t$ and its corresponding human pose $\mathbf{p}_t$, and force it to generate frame $\mathbf{x}_{t+n}$ given pose $\mathbf{p}_{t+n}$, where n is defined randomly for every iteration at training time. Training to jump to random frames in time gives our network a clear signal the task at hand due to the large pixel difference between frames far apart in time.

To train our network, we use the compound loss from Dosovitskiy & Brox (2016). Our network is optimized to minimize the objective given by

$$\mathcal{L} = \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{feat}} + \mathcal{L}_{\text{Gen}}, \tag{5}$$

where $\mathcal{L}_{\text{img}}$ is the loss in image space defined by

$$\mathcal{L}_{\text{img}} = \|\mathbf{x}_{t+n} - \hat{\mathbf{x}}_{t+n}\|_2^2, \tag{6}$$

where $\mathbf{x}_{t+n}$ and $\hat{\mathbf{x}}_{t+n}$ are the target and predicted frames, respectively. The image loss intuitively guides our network towards a rough blurry pixel-leven frame prediction that reflects most details of the target image. $\mathcal{L}_{\text{feat}}$ is the loss in feature space define by

$$\begin{aligned}\mathcal{L}_{\text{feat}} = &\|C_1\left(\mathbf{x}_{t+n}\right) - C_1\left(\hat{\mathbf{x}}_{t+n}\right)\|_2^2 \\ &+ \|C_2\left(\mathbf{x}_{t+n}\right) - C_2\left(\hat{\mathbf{x}}_{t+n}\right)\|_2^2,\end{aligned} \tag{7}$$

where $C_1\left(.\right)$ extracts features representing mostly image appearance, and $C_2\left(.\right)$ extracts features representing mostly image structure. Combining appearance sensitive features with structure sensitive features gives our network a learning signal that allows it to make frame predictions with accurate appearance while also enforcing correct structure. $\mathcal{L}_{\text{Gen}}$ is the term in adversarial loss that allows our model to generate realistic looking images and is defined by

$$\mathcal{L}_{\text{Gen}} = -\log D\left(\left[\mathbf{p}_{t+n}, \hat{\mathbf{x}}_{t+n}\right]\right), \tag{8}$$

where $\mathbf{x}_{t+n}$ is the target image, $\mathbf{p}_{t+n}$ is the human pose corresponding to the target image, and $D\left(.\right)$ is the discriminator network in adversarial loss. This sub-loss allows our network to generate images that reflect a similar level of detail as the images observed in the training data.

During the optimization of D, we use the mismatch term proposed by Reed et al. (2016b), which allows the discriminator D to become sensitive to mismatch between the generation and the condition. The discriminator loss is defined by

$$\begin{aligned}\mathcal{L}_{\text{Disc}} = &-\log D\left(\left[\mathbf{p}_{t+n}, \mathbf{x}_{t+n}\right]\right) \\ &- 0.5\log\left(1 - D\left(\left[\mathbf{p}_{t+n}, \hat{\mathbf{x}}_{t+n}\right]\right)\right) \\ &- 0.5\log\left(1 - D\left(\left[\mathbf{p}_{t+n}, \mathbf{x}_{t}\right]\right)\right),\end{aligned} \tag{9}$$

while optimizing our generator with respect to the adversarial loss, the mismatch-aware term sends a stronger signal to our generator resulting in higher quality image generation, and network optimization. Essentially, having a discriminator that knows the correct structure-image relationship, reduces the parameter search space of our generator while optimizing to fool the discriminator into believing the generated image is real. The latter in combination with the rest of loss terms allows our network to produce high quality image generation given the structure condition.

6. Experiments

In this section, we present experiments on pixel-level video prediction of human actions on the Penn Action (Weiyu Zhang & Derpanis, 2013) and Human 3.6M datasets (Ionescu et al., 2014). Pose landmarks and video frames are normalized to be between -1 and 1, and frames are cropped based on temporal tubes to remove as much background as possible while making sure the human of interest is in all frames. For the feature similarity loss term (Equation 7), we use we use the last convolutional layer in AlexNet (Krizhevsky et al., 2012) as C_1, and the last layer of the Hourglass Network in Newell et al. (2016) as C_2. We augmented the available video data by performing horizontal flips randomly at training time for Penn Action. Motion-based pixel-level quantitative evaluation using Peak Signal-to-Noise Ratio (PSNR), analysis, and control experiments can be found in the supplementary material. For video illustration of our method, please refer to the project website: `https://sites.google.com/a/umich.edu/rubenevillegas/hierch_vid`.

We compare our method against two baselines based on convolutional LSTM and optical flow. A convolutional LSTM baseline (Shi et al., 2015) was trained with adversarial loss (Mathieu et al., 2016) and the feature similarity loss (Equation 7). An optical flow based baseline used the last observed optical flow (Farneback, 2003) to move the pixels of the last observed frame into the future.

We follow a human psycho-physical quantitative evaluation metric similar to Vondrick et al. (2016). Amazon Mechanical Turk (AMT) workers are given a two-alternative choice to indicate which of two videos looks more realistic. Specifically, the workers are shown a pair of videos (generated by two different methods) consisting of the same input frames indicated by a green box and predicted frames indicated by a red box, in addition to the action label of the video. The workers are instructed to make their decision based on the frames in the red box. Additionally, we train a Two-stream action recognition network (Simonyan & Zisserman, 2014) on the Penn Action dataset and test on the generated videos to evaluate if our network is able to generate videos predicting the activities observed in the original dataset. We do not perform action classification experiments on the

Method	Temporal Stream	Spatial Stream	Combined
Real Test Data *	66.6%	63.3%	72.1%
Ours	**35.7**%	**52.7**%	**59.0**%
Convolutional LSTM	13.9%	45.1%	46.4%
Optical Flow	13.9%	39.2%	34.9%

Table 1. Activity recognition evaluation.

"Which video is more realistic?"	Baseball	Clean & jerk	Golf	Jumping jacks	Jump rope	Tennis	Mean
Prefers ours over Convolutional LSTM	89.5%	87.2%	84.7%	83.0%	66.7%	88.2%	82.4%
Prefers ours over Optical Flow	87.8%	86.5%	80.3%	88.9%	86.2%	85.6%	86.1%

Table 2. **Penn Action Video Generation Preference:** We show videos from two methods to Amazon Mechanical Turk workers and ask them to indicate which is more realistic. The table shows the percentage of times workers preferred our model against baselines. A majority of the time workers prefer predictions from our model. We merged baseball pitch and baseball swing into baseball, and tennis forehand and tennis serve into tennis.

Human3.6M dataset due to high uncertainty in the human movements and high motion similarity amongst actions.

Architectures. The sequence prediction LSTM is made of a single layer encoder-decoder LSTM with tied parameters, 1024 hidden units, and tanh output activation. Note that the decoder LSTM does not observe any inputs other than the hidden units from the encoder LSTM as initial hidden units. The image and pose encoders are built with the same architecture as VGG16 (Simonyan & Zisserman, 2015) up to the pooling layer, except that the pose encoder takes in the pose heat-maps as an image made of L channels, and the image encoder takes a regular 3-channel image. The decoder is the mirrored architecture of the image encoder where we perform unpooling followed by deconvolution, and a final tanh activation. The convolutional LSTM baseline is built with the same architecture as the image encoder and decoder, but there is a convolutional LSTM layer with the same kernel size and number of channels as the last layer in the image encoder connecting them.

6.1. Penn Action Dataset

Experimental setting. The Penn Action dataset is composed of 2326 video sequences of 15 different actions and 13 human joint annotations for each sequence. To train our image generator, we use the standard train split provided in the dataset. To train our pose predictor, we sub-sample the actions in the standard train-test split due to very noisy joint ground-truth. We used videos from the actions of baseball pitch, baseball swing, clean and jerk, golf swing, jumping jacks, jump rope, tennis forehand, and tennis serve. Our pose predictor is trained to observe 10 inputs and predict 32 steps, and tested on predicting up to 64 steps (some videos' groundtruth end before 64 steps). Our image generator is trained to make single random jumps within 30 steps into the future. Our evaluations are performed on a single clip that starts at the first frame of each video.

AMT results. These experiments were performed by 66 unique workers, where a total of 1848 comparisons were made (934 against convolutional LSTM and 914 against optical flow baseline). As shown in Table 2 and Figure 5, our method is capable of generating more realistic sequences compared to the baselines. Quantitatively, the action sequences generated by our network are perceptually higher quality than the baselines and also predict the correct action sequence. A relatively small (although still substantial) margin is observed when comparing to convolutional LSTM for the jump rope action (i.e., 66.7% for ours vs 33.3% for Convolutional LSTM). We hypothesize that convolutional LSTM is able to do a reasonable job for this action class due the highly cyclic motion nature of jumping up and down in place. The remainder of the human actions contain more complicated non-linear motion, which is much more complicated to predict. Overall, our method outperforms the baselines by a large margin (i.e. 82.4% for ours vs 17.6% for Convolutional LSTM, and 86.1% for ours vs 13.9% for Optical Flow). Side by side video comparison for all actions can be found in our project website.

Action recognition results. To see whether the generated videos contain actions that can fool a CNN trained for action recognition, we train a Two-Stream CNN on the PennAction dataset. In Table 1, "Temporal Stream" denotes the network that observes motion as concatenated optical flow (Farneback's optical flow) images as input, and "Spatial Stream" denotes the network that observes single image as input. "Combined" denotes the averaging of the output probability vectors from the Temporal and Spatial stream. "Real test data" denotes evaluation on the ground-truth videos (i.e. perfect prediction).

From Table 1, it is shown that our network is able to generate videos that are far more representative of the correct action compared to all baselines, in both Temporal and Spatial stream, regardless of using a neural network as the judge. When combining both Temporal and Spatial streams, our network achieves the best quality videos in terms of making a pixel-level prediction of the correct action.

Pixel-level evaluation and control experiments. We evaluate the frames generated by our method using PSNR

"Which video is more realistic?"	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing
Prefers ours over Convolutional LSTM	67.6%	75.9%	74.7%	79.5%	69.7%	66.2%	69.7%
Prefers ours over Optical Flow	61.4%	89.3%	43.8%	80.3%	84.5%	52.0%	75.3%

"Which video is more realistic?"	Purchases	Sitting	Sittingdown	Smoking	Waiting	Walking	Mean
Prefers ours over Convolutional LSTM	79.0%	38.0%	54.7%	70.4%	50.0%	86.0%	70.3%
Prefers ours over Optical Flow	85.7%	35.1%	46.7%	73.3%	84.3%	90.8%	72.3%

Table 3. **Human3.6M Video Generation Preference:** We show videos from two methods to Amazon Mechanical Turk workers and ask them to indicate which is more realistic. The table shows the percentage of times workers preferred our model against baselines. Most of the time workers prefer predictions from our model. We merge baseball pitch and baseball swing into baseball, and tennis forehand and tennis serve into tennis.

as measure, and separated the test data based on amount of motion, as suggested by Villegas et al. (2017). From these experiments, we conclude that pixel-level evaluation highly depends on predicting the exact future observed in the ground-truth. Highest PSNR scores are achieved when trajectories of the exact future is used to generate the future frames. Due to space constraints, we ask the reader to please refer to the supplementary material for more detailed quantitative and qualitative analysis.

6.2. Human3.6M Dataset

Experimental settings. The Human3.6M dataset (Ionescu et al., 2014) is composed of 3.6 million 3D human poses (we use the provided 2D pose projections) composed of 32 joints and corresponding images taken from 11 professional actors in 17 scenarios. For training, we use subjects number 1, 5, 6, 7, and 8, and test on subjects number 9 and 11. Our pose predictor is trained to observe 10 inputs and predict 64 steps, and tested on predicting 128 steps. Our image generator is trained to make single random jumps anywhere in the training videos. We evaluate on a single clip from each test video that starts at the exact middle of the video to make sure there is motion occurring.

AMT results. We collected a total of 2203 comparisons (1086 against convolutional LSTM and 1117 against optical flow baseline) from 71 unique workers. As shown in Table 3, the videos generated by our network are perceptually higher quality and reflect a reasonable future compared to the baselines on average. Unexpectedly, our network does not perform well on videos where the action involves minimal motion, such as sitting, sitting down, eating, taking a photo, and waiting. These actions usually involve the person staying still or making very unnoticeable motion which can result in a static prediction (by convolutional LSTM and/or optical flow) making frames look far more realistic than the prediction from our network. Overall, our method outperforms the baselines by a large margin (i.e. 70.3% for ours vs 29.7% for Convolutional LSTM, and 72.3% for ours vs 27.7% for Optical Flow). Figure 5 shows that our network generates far higher quality future frames compared to the convolutional LSTM baseline. Side by side video comparison for all actions can be found in our project website.

Pixel-level evaluation and control experiments. Following the same procedure as Section 6.1, we evaluated the predicted videos using PSNR and separated the test data by motion. Due to the high uncertainty and number of prediction steps in these videos, the predicted future can largely deviate from the exact future observed in the ground-truth. The highest PSNR scores are again achieved when the exact future pose is used to generate the video frames; however, there is an even larger gap compared to the results in Section 6.1. Due to space constraints, we ask the reader to please refer to the supplementary material for more detailed quantitative and qualitative analysis.

7. Conclusion and Future Work

We propose a hierarchical approach of pixel-level video prediction. Using human action videos as benchmark, we have demonstrated that our hierarchical prediction approach is able to predict up to 128 future frames, which is an order of magnitude improvement in terms of effective temporal scale of the prediction.

The success of our approach demonstrates that it can be greatly beneficial to incorporate the proper high-level structure into the generative process. At the same time, an important open research question would be how to automatically learn such structures without domain knowledge. We leave this as future work.

Another limitation of this work is that it generates a single future trajectory. For an agent to make a better estimation of what the future looks like, we would need more than one generated future. Future work will involve the generation of many futures given using a probabilistic sequence model.

Finally, our model does not handle background motion. This is a highly challenging task since background comes in and out of sight. Predicting background motion will require a generative model that hallucinates the unseen background. We also leave this as future work.

Acknowledgments

This work was supported in part by ONR N00014-13-1-0762, NSF CAREER IIS-1453651, Gift from Bosch Research, and Sloan Research Fellowship. We thank NVIDIA for donating K40c and TITAN X GPUs.

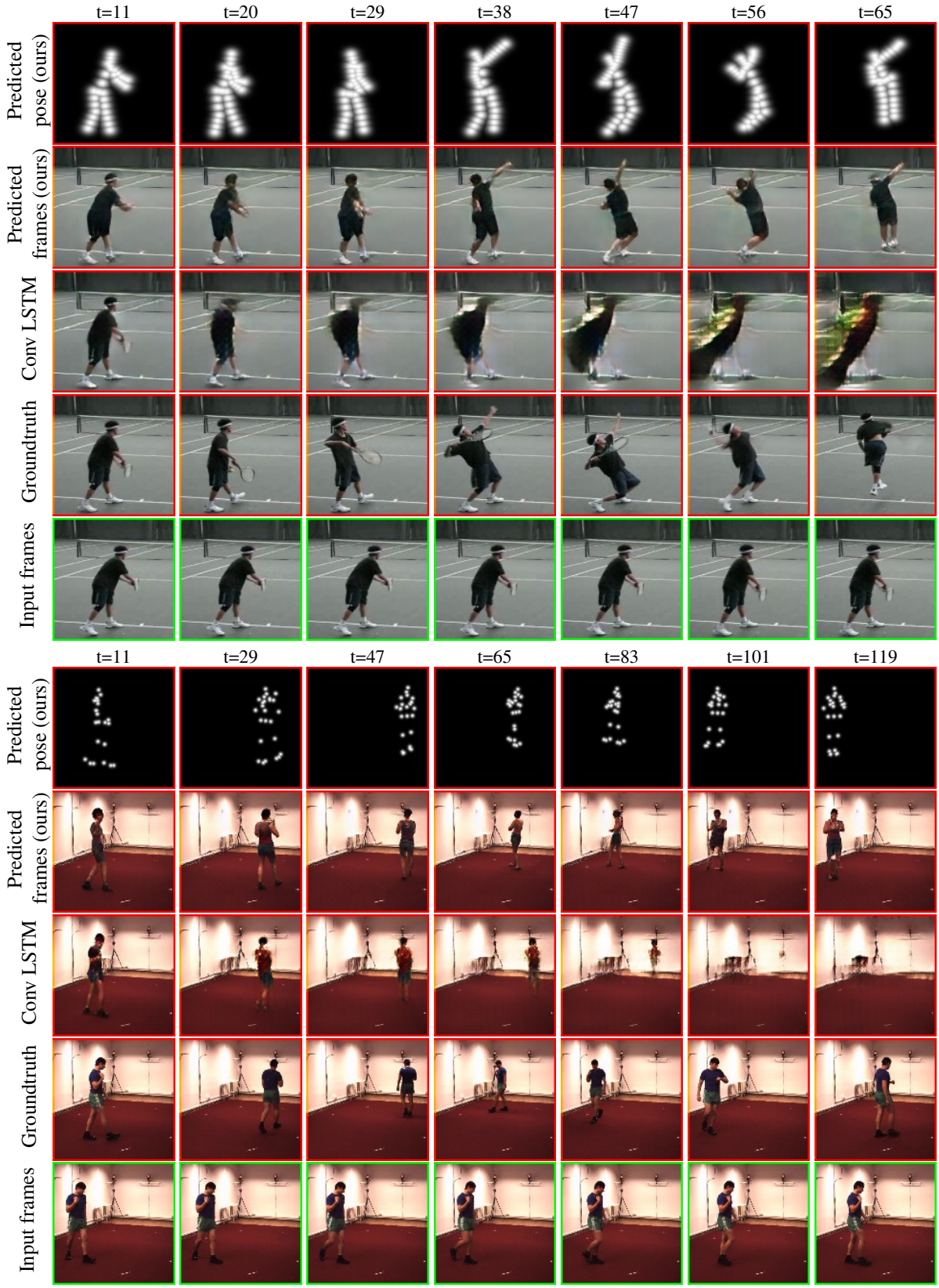

Figure 5. Qualitative evaluation of our network for 55 step prediction on Penn Action (top rows), and 109 step prediction on Human3.6M (bottom rows). Our algorithm observes 10 previous input frames, estimates the human pose, predicts the pose sequence of the future, and it finally generates the future frames. Green box denotes input and red box denotes prediction. We show the last 7 input frames. Side by side video comparisons can be found in our project website.

References

Dosovitskiy, A. and Brox, T. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016. 5

Farneback, G. Two-frame motion estimation based on polynomial expansion. In *SCIA*, 2003. 5

Finn, C., Goodfellow, I. J., and Levine, S. Unsupervised learning for physical interaction through video prediction. In *NIPS*. 2016. 1, 2

Fragkiadaki, K., Levine, S., Felsen, P., and Malik, J. Recurrent network models for human dynamics. In *ICCV*, 2015. 1

Goroshin, R., Mathieu, M., and LeCun, Y. Learning to linearize under uncertainty. In *NIPS*. 2015. 1

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. 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, 7

Jayaraman, D. and Grauman, K. Learning image representations tied to ego-motion. In *ICCV*. 2015. 1

Jayaraman, D. and Grauman, K. Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion. *arXiv preprint:1605.00164*, 2016. 1

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012. 1, 5

Lee, N. *Modeling of Dynamic Environments for Visual Forecasting of American Football Plays*. PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2015. 1

Lotter, W., Kreiman, G., and Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*. 2017. 1, 2

Mathieu, M., Couprie, C., and LeCun, Y. Deep multi-scale video prediction beyond mean square error. In *ICLR*. 2016. 1, 2, 5

Michalski, V., Memisevic, R., and Konda, K. Modeling deep temporal dependencies with recurrent "grammar cells". In *NIPS*, 2014. 2

Mittelman, R., Kuipers, B., Savarese, S., and Lee, H. Structured recurrent temporal restricted boltzmann machines. In *ICML*. 2014. 2

Newell, A., Yang, K., and Deng, J. Stacked hourglass networks for human pose estimation. In *ECCV*. 2016. 2, 4, 5

Oh, J., Guo, X., Lee, H., Lewis, R. L., and Singh, S. Action-conditional video prediction using deep networks in atari games. In *NIPS*. 2015. 1, 2

Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint:1412.6604*, 2014. 1, 2

Reed, S., Zhang, Y., Zhang, Y., and Lee, H. Deep visual analogy-making. In *NIPS*. 2015. 2, 3

Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. Learning what and where to draw. In *NIPS*, 2016a. 2

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text-to-image synthesis. In *ICML*. 2016b. 5

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and WOO, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*. 2015. 5

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *NIPS*. 2014. 5

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

Srivastava, N., Mansimov, E., and Salakhudinov, R. Unsupervised learning of video representations using lstms. In *ICML*. 2015. 1, 2

Sutskever, I., Hinton, G. E., and Taylor, G. W. The recurrent temporal restricted boltzmann machine. In *NIPS*. 2009. 2

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *CVPR*. 2015. 1

Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. Decomposing motion and content for natural video sequence prediction. In *ICLR*. 2017. 1, 2, 7

Vondrick, C., Pirsiavash, H., and Torralba, A. Generating videos with scene dynamics. In *NIPS*. 2016. 2, 5

Walker, J., Gupta , A., and Hebert , M. Patch to the future: Unsupervised visual prediction. In *CVPR*, 2014. 1

Weiyu Zhang, M. Z. and Derpanis, K. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*. 2013. 5

Xue, T., Wu, J., Bouman, K. L., and Freeman, W. T. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *NIPS*, 2016. 2

Yuen, J. and Torralba, A. A data-driven approach for event prediction. In *ECCV*. 2010. 1