DECOMPOSING MOTION AND CONTENT FOR NATURAL VIDEO SEQUENCE PREDICTION

Ruben Villegas[1] **Jimei Yang**[2] **Seunghoon Hong**[3,*] **Xunyu Lin**[4,*] **Honglak Lee**[1,5]
[1]University of Michigan, Ann Arbor, USA
[2]Adobe Research
[3]POSTECH, Pohang, Korea
[4]Beihang University, Beijing, China
[5]Google Brain, Mountain View, CA 94043

ABSTRACT

We propose a deep neural network for the prediction of future frames in natural video sequences. To effectively handle complex evolution of pixels in videos, we propose to decompose the motion and content, two key components generating dynamics in videos. Our model is built upon the Encoder-Decoder Convolutional Neural Network and Convolutional LSTM for pixel-level prediction, which independently capture the spatial layout of an image and the corresponding temporal dynamics. By independently modeling motion and content, predicting the next frame reduces to converting the extracted content features into the next frame content by the identified motion features, which simplifies the task of prediction. Our model is end-to-end trainable over multiple time steps, and naturally learns to decompose motion and content without separate training. We evaluate the proposed network architecture on human activity videos using KTH, Weizmann action, and UCF-101 datasets. We show state-of-the-art performance in comparison to recent approaches. To the best of our knowledge, this is the first end-to-end trainable network architecture with motion and content separation to model the spatio-temporal dynamics for pixel-level future prediction in natural videos.

1 INTRODUCTION

Understanding videos has been one of the most important tasks in the field of computer vision. Compared to still images, the temporal component of videos provides much richer descriptions of the visual world, such as interaction between objects, human activities, and so on. Amongst the various tasks applicable on videos, the task of anticipating the future has recently received increased attention in the research community. Most prior works in this direction focus on predicting high-level semantics in a video such as action (Vondrick et al., 2015; Ryoo, 2011; Lan et al., 2014), event (Yuen and Torralba, 2010; Hoai and Torre, 2013) and motion (Pintea et al., 2014; Walker et al., 2014; Pickup et al., 2014; Walker et al., 2016). Forecasting semantics provides information about *what will happen* in a video, and is essential to automate decision making. However, the predicted semantics are often specific to a particular task and provide only a partial description of the future. Also, training such models often requires heavily labeled training data which leads to tremendous annotation costs especially with videos.

In this work, we aim to address the problem of prediction of future frames in natural video sequences. Pixel-level predictions provide dense and direct description of the visual world, and existing video recognition models can be adopted on top of the predicted frames to infer various semantics of the future. Spatio-temporal correlations in videos provide a self-supervision for frame prediction, which enables purely unsupervised training of a model by observing raw video frames. Unfortunately, estimating frames is an extremely challenging task, not only because of the inherent uncertainty of the future, but also various factors of variation in videos leading to complicated dynamics in raw pixel values. There have been a number of recent attempts on frame prediction (Srivastava et al., 2015; Mathieu et al., 2015; Oh et al., 2015; Goroshin et al., 2015; Lotter et al., 2015; Ranzato et al., 2014),

[*]This work was done while SH and XL were visiting the University of Michigan.

which use a single encoder that needs to reason about all the different variations occurring in videos in order to make predictions of the future, or require extra information like foreground-background segmentation masks and static background (Vondrick et al., 2016).

We propose a Motion-Content Network (MCnet) for robust future frame prediction. Our intuition is to split the inputs for video prediction into two easily identifiable groups, motion and content, and independently capture each information stream with separate encoder pathways. In this architecture, the *motion* pathway encodes the local dynamics of spatial regions, while the *content* pathway encodes the spatial layout of the salient parts of an image. The prediction of the future frame is then achieved by transforming the content of the last observed frame given the identified dynamics up to the last observation. Somewhat surprisingly, we show that such a network is end-to-end trainable *without individual path way supervision*. Specifically, we show that an asymmetric architecture for the two pathways enables such decompositions without explicit supervision. The contributions of this paper are summarized below.

- We propose MCnet for the task of frame prediction, which separates the information streams (motion and content) into different encoder pathways.
- The proposed network is end-to-end trainable and naturally learns to decompose motion and content without separate training, and reduces the task of frame prediction to transforming the last observed frame into the next by the observed motion.
- We evaluate the proposed model on challenging real-world video datasets, and show that it outperforms previous approaches on frame prediction.

The rest of the paper is organized as follows. We briefly review related work in Section 2, and introduce an overview of the proposed algorithm in Section 3. The detailed configuration of the proposed network is described in Section 4. Section 5 describes training and inference procedure. Section 6 illustrates implementation details and experimental results on challenging benchmarks.

2 Related work

The problem of visual future prediction has received growing interests in the computer vision community. It has led to various tasks depending on the objective of future prediction, such as human activity (Vondrick et al., 2015; Ryoo, 2011; Lan et al., 2014), event (Yuen and Torralba, 2010; Hoai and Torre, 2013) and geometric path (Walker et al., 2014). Although they achieved reasonable success in specific tasks, they are often limited to estimating predefined semantics, and require fully-labeled training data. To alleviate this issue, approaches predicting representation of the future beyond semantic labels have been proposed. Walker et al. (2014) proposed a data-driven approach to predict motion of a moving object, and coarse hallucination of the predicted motion. Vondrick et al. (2015) proposed a deep regression network to predict feature representations of the future frames. These approaches are supervised, and provide coarse predictions of how the future would look like. Our work also focuses on unsupervised learning for prediction of the future but to a more direct visual prediction task, frame prediction.

Compared to predicting semantics, pixel-level prediction has been less investigated due to the difficulties in modeling evolution of raw pixels over time. Fortunately, recent advances in deep learning provide a powerful tool for sequence modeling, and enable the creation of novel architectures for modeling complex sequential data. Ranzato et al. (2014) applied a recurrent neural network developed for language modeling to frame prediction by posing the task as classification of each image region to one of quantized patch dictionaries. Srivastava et al. (2015) applied a sequence-to-sequence model to video prediction, and showed that Long Short-Term Memory (LSTM) is able to capture pixel dynamics. Oh et al. (2015) proposed an action-conditional encoder-decoder network to predict future frames in Atari games. Besides the different choice of architecture, some other works addressed the importance of selecting right objective function: Lotter et al. (2015) used adversarial loss with combined CNN and LSTM architectures, and Mathieu et al. (2015) employed similar adversarial loss with additional regularization using a multi-scale encoder-decoder network. Vondrick et al. (2016) proposed a generative adversarial network for video which by generating a background-foreground mask is able to generate realistic-looking video sequences. However, none of the previously mentioned approaches exploit spatial and temporal information separately in their network architectures. The closest work to ours is Xue et al. (2016) in terms of the input. The

differences are (1) Our model is deterministic and theirs is probabilistic, (2) our motion encoder is based on convolutional LSTM (Shi et al., 2015) which is a more natural module to model long-term dynamics, (3) our content encoder observes a single scale input and theirs observes many scales, and (4) we directly generate image pixels values which is a more complicated task. We aim to exploit the existing spatio-temporal correlations in videos by decomposing the motion and content in our network architecture.

To the best of our knowledge, the idea of separating motion and content has not been investigated in the task of deterministic frame prediction. The proposed architecture shares similarities to the two-stream CNN (Simonyan and Zisserman, 2014), which is designed for action recognition to jointly exploit the information from frames and their temporal dynamics. However, in contrast to their network we aim to learn features for temporal dynamics directly from the raw pixels, and we use the identified features from the motion in combination with spatial features to make pixel-level predictions of the future.

3 Algorithm Overview

In this section, we formally define the task of frame prediction and the role of each component in the proposed architecture. Let $\mathbf{x}_t \in \mathrm{R}^{w \times h \times c}$ denote the t-th frame in an input video $\mathbf{x}$, where w, h and c denote width, height and number of channels, respectively. The objective of frame prediction is to generate the future frame $\hat{\mathbf{x}}_{t+1}$ given the input frames $\mathbf{x}_{1:t}$.

At the t-th time step, our network observes a history of previous consecutive frames up to frame t, and generates the prediction of the next frame $\hat{\mathbf{x}}_{t+1}$ as follows:

- **Motion encoder** recurrently takes an image difference input between frame $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ starting from $t = 2$, and produces the hidden representation $\mathbf{d}_t$ encoding the temporal dynamics of the scene components (Section 4.1).
- **Content encoder** takes the last observed frame $\mathbf{x}_t$ as an input, and outputs the hidden representation $\mathbf{s}_t$ that encodes the spatial layout of the scene (Section 4.2).
- **Multi-scale Motion-Content Residual** takes the computed features from both, the motion and content encoders, at every scale right before pooling and computes residuals $\mathbf{r}_t$ (He et al., 2015) to aid the information loss caused by pooling in the encoding phase (Section 4.3).
- **Combination Layers and Decoder** takes the outputs from both encoder pathways and residual connections, $\mathbf{d}_t$, $\mathbf{s}_t$ and $\mathbf{r}_t$, and combines them to produce a pixel-level prediction of the next frame $\hat{\mathbf{x}}_{t+1}$ (Section 4.4).

The overall architecture of the proposed algorithm is described in Figure 1. The prediction of multiple frames, $\hat{\mathbf{x}}_{t+1:t+T}$, can be achieved by recursively performing the above procedures over T time steps (Section 5). In the following section, we describe each component in the proposed architecture.

4 Architecture

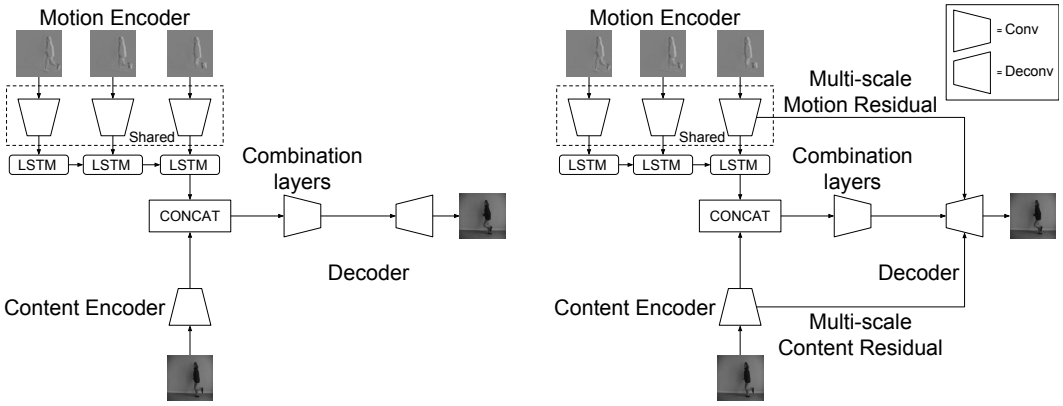
This section describes the detailed configuration of the proposed architecture, including the two encoder pathways, multi-scale residual connections, combination layers, and decoder.

4.1 Motion Encoder

The motion encoder captures the temporal dynamics of the scene's components by recurrently observing subsequent difference image computed from $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$, and outputs motion features by

$$[\mathbf{d}_t, \mathbf{c}_t] = f^{\text{dyn}}\left(\mathbf{x}_t - \mathbf{x}_{t-1}, \mathbf{d}_{t-1}, \mathbf{c}_{t-1}\right), \tag{1}$$

where $\mathbf{x}_t - \mathbf{x}_{t-1}$ denotes element-wise subtraction between frames at time t and $t-1$, $\mathbf{d}_t \in \mathbb{R}^{w' \times h' \times c'}$ is the feature tensor encoding the motion across the observed difference image inputs, and $\mathbf{c}_t \in \mathbb{R}^{w' \times h' \times c'}$ is a memory cell that retains information of the dynamics observed through time. f^{dyn} is implemented in a fully-convolutional way to allow our model to identify local dynamics of frames rather than complicated global motion. For this, we use an encoder CNN with a Convolutional LSTM (Shi et al., 2015) layer on top.



(a) Base MCnet (b) MCnet with Multi-scale Motion-Content Residuals

Figure 1: Overall architecture of the proposed network. (a) Illustrates MCnet without the Motion-Content Residual *skip connections*, and (b) Illustrates MCnet with such connections. Our network observes a history of image differences through the motion encoder and last observed image through the content encoder. Subsequently, our network proceeds to compute motion-content features and communicates them to the decoder for the prediction of the next frame.

4.2 Content Encoder

The content encoder extracts important spatial features from a single frame, such as the spatial layout of the scene and salient objects in a video. Specifically, it takes the last observed frame $\mathbf{x}_t$ as an input, and produces content features by

$$\mathbf{s}_t = f^{\text{cont}}\left(\mathbf{x}_t\right), \tag{2}$$

where $\mathbf{s}_t \in \mathbb{R}^{w' \times h' \times c'}$ is the feature encoding the spatial content of the last observed frame, and f^{cont} is implemented by a Convolutional Neural Network (CNN) that specializes on extracting features from single frame.

It is important to note that our model employs an *asymmetric* architecture for the motion and content encoder. The content encoder takes the last observed frame, which keeps the most critical clue to reconstruct spatial layout of near future, but has no information about dynamics. On the other hand, the motion encoder takes a history of previous image differences, which are less informative to guess the future spatial layout compared to the last observed frame, yet contain important spatio-temporal variations occurring over time. This asymmetric architecture encourages encoders to exploit each of two critical information to predict the future–content and motion–individually, and enables the model to learn motion and contents decomposition naturally without any supervision.

4.3 Multi-scale Motion-Content Residual

To prevent information loss after the pooling operations in our motion and content encoders, we make use of residual connections (He et al., 2015). The residual connections in our network communicate motion-content features at every scale into the decoder layers after unpooling operations. The residual feature at layer l is computed by

$$\mathbf{r}_t^l = f^{\text{res}}\left(\left[\mathbf{s}_t^l, \mathbf{d}_t^l\right]\right)^l, \tag{3}$$

where $\mathbf{r}_t^l$ is the residual output at layer l, $\left[\mathbf{s}_t^l, \mathbf{d}_t^l\right]$ is the concatenation of the motion and content features along the depth dimension at layer l of their respective encoders, $f^{\text{res}}\left(.\right)^l$ the residual function at layer l implemented as consecutive convolution layers and rectification with a final linear layer.

4.4 Combination Layers and Decoder

The outputs from the two encoder pathways, $\mathbf{d}_t$ and $\mathbf{s}_t$, encode a high-level representation of motion and content, respectively. Given these representations, the objective of the decoder is to generate a pixel-level prediction of the next frame $\hat{\mathbf{x}}_{t+1} \in \mathbb{R}^{w \times h \times c}$. To this end, it first combines the motion and content back into a unified representation by

$$\mathbf{f}_t = g^{\text{comb}}\left(\left[\mathbf{d}_t, \mathbf{s}_t\right]\right), \tag{4}$$

where $[\mathbf{d}_t, \mathbf{s}_t] \in \mathbb{R}^{w' \times h' \times 2c'}$ denotes the concatenation of the higher-level motion and content features in the depth dimension, and $\mathbf{f}_t \in \mathbb{R}^{w' \times h' \times c'}$ denotes the combined high-level representation of motion and content. g^{comb} is implemented by a CNN with bottleneck layers (Hinton and Salakhutdinov, 2006); it first projects both $\mathbf{d}_t$ and $\mathbf{s}_t$ into a lower-dimensional embedding space, and then puts it back to the original size to construct the combined feature $\mathbf{f}_t$. Intuitively, $\mathbf{f}_t$ can be viewed as the content feature of the next time step, $\mathbf{s}_{t+1}$, which is generated by transforming $\mathbf{s}_t$ using the observed dynamics encoded in $\mathbf{d}_t$. Then our decoder places $\mathbf{f}_t$ back into the original pixel space by

$$\hat{\mathbf{x}}_{t+1} = g^{\text{dec}}\left(\mathbf{f}_t, \mathbf{r}_t\right). \tag{5}$$

where $\mathbf{r}_t$ are the residual connections from every layer of the motion and content encoders before pooling sent to every layer of the decoder after unpooling. We employ the deconvolution network (Zeiler et al., 2011) for our decoder network g^{dec}, which is composed of multiple successive operations of deconvolution, rectification and unpooling with the addition of the motion-content residual connections after each unpooling operation. The output layer is passed through a $\tanh\left(.\right)$ activation function. Unpooling with fixed switches are used to upsample the intermediate activation maps.

5 Inference and Training

The above descriptions up to Section 4 are about the procedures for single frame prediction. This section presents the extension of our algorithm for the prediction of multiple time steps.

5.1 Multi-step prediction

Given an input video, our network observes the first n frames as image difference between frame $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ starting from $t = 2$ up to $t = n$ to encode initial temporal dynamics through the motion encoder, and the last frame $\mathbf{x}_n$ is given to the content encoder to be transformed into the first prediction $\hat{\mathbf{x}}_{t+1}$ by the identified motion features.

Then for each time step $t \in [n+1, n+T]$ where T is desired number of prediction steps, our network takes the difference image between the first prediction $\hat{\mathbf{x}}_{t+1}$ and the previous image $\mathbf{x}_t$, and the first prediction $\hat{\mathbf{x}}_{t+1}$ itself to predict the next frame $\hat{\mathbf{x}}_{t+2}$, and so on.

5.2 Training Objective

To train our network, we use an objective function composed of different sub-losses similar to Mathieu et al. (2015). Given the training data $D = \{\mathbf{x}^{(i)}_{1,\ldots,T}\}_{i=1}^{N}$, our model is trained to minimize the prediction loss by

$$\mathcal{L} = \alpha \mathcal{L}_{\text{img}} + \beta \mathcal{L}_{\text{GAN}} \tag{6}$$

where α and β are hyper-parameters that control the effect of each sub-loss during optimization. $\mathcal{L}_{\text{img}}$ is the loss in image space from Mathieu et al. (2015) defined by

$$\mathcal{L}_{\text{img}} = \mathcal{L}_p\left(\mathbf{x}_{t+k}, \hat{\mathbf{x}}_{t+k}\right) + \mathcal{L}_{gdl}\left(\mathbf{x}_{t+k}, \hat{\mathbf{x}}_{t+k}\right), \tag{7}$$

$$\text{where} \quad \mathcal{L}_p\left(\mathbf{y}, \mathbf{z}\right) = \sum_{k=1}^{T} ||\mathbf{y} - \mathbf{z}||_p^p, \tag{8}$$

$$\mathcal{L}_{gdl}\left(\mathbf{y}, \mathbf{z}\right) = \sum_{i,j}^{h,w} |\left(|\mathbf{y}_{i,j} - \mathbf{y}_{i-1,j}| - |\mathbf{z}_{i,j} - \mathbf{z}_{i-1,j}|\right)|^{\lambda} + |\left(|\mathbf{y}_{i,j-1} - \mathbf{y}_{i,j}| - |\mathbf{z}_{i,j-1} - \mathbf{z}_{i,j}|\right)|^{\lambda}. \tag{9}$$

Here, $\mathbf{x}_{t+k}$ and $\hat{\mathbf{x}}_{t+k}$ are the target and predicted frames, respectively, and λ and p are hyper-parameters for each $\mathcal{L}_p$ and $\mathcal{L}_{gdl}$. Intuitively, $\mathcal{L}_p$ guides our network to match the average pixel values directly, while $\mathcal{L}_{gdl}$ guides our network to match the gradients of such pixel values. Overall, $\mathcal{L}_{\text{img}}$ guides our network to learn parameters towards generating the correct average sequence given the input. Training to generate average sequences, however, results in somewhat blurry generations which is the reason we use an additional sub-loss. $\mathcal{L}_{\text{GAN}}$ is the generator loss in adversarial training to allow our model to predict realistic looking frames and it is defined by

$$\mathcal{L}_{\text{GAN}} = -\log D\left(\left[\mathbf{x}_{1:t}, G\left(\mathbf{x}_{1:t}\right)\right]\right), \tag{10}$$

where $\mathbf{x}_{1:t}$ is the concatenation of the input images, $\mathbf{x}_{t+1:t+T}$ is the concatenation of the ground-truth future images, $G(\mathbf{x}_{1:t}) = \hat{\mathbf{x}}_{t+1:t+T}$ is the concatenation of all predicted images, all along depth dimension, and $D(.)$ is the discriminator in adversarial training. The discriminative loss in adversarial training is defined by

$$\mathcal{L}_{\text{disc}} = -\log D([\mathbf{x}_{1:t}, \mathbf{x}_{t+1:t+T}]) - \log\left(1 - D([\mathbf{x}_{1:t}, G(\mathbf{x}_{1:t})])\right), \tag{11}$$

$\mathcal{L}_{\text{GAN}}$ in addition to $\mathcal{L}_{\text{img}}$ allows our network to not only generate the target sequence, but at the same time enforce realism in the images through visual sharpness that fools the human eye. Note that our model uses its predictions as an input for the next time-step during the training; it enables the gradients to flow through time, and makes the network robust for error propagation during prediction. For more a detailed description about adversarial training, please refer to Appendix D.

6 EXPERIMENTS

This section we present experiments using our network for video generation. We first evaluate our network, MCnet on the KTH (Schuldt et al., 2004) and Weizmann action (Gorelick et al., 2007) datasets, and compare against an baseline convolutional LSTM (ConvLSTM) (Shi et al., 2015). We then proceed to evaluate on the more challenging UCF-101 (Soomro et al., 2012) dataset, in which we compare against the same ConvLSTM baseline and also the current state-of-the-art method by Mathieu et al. (2015). For all our experiments we use $\alpha = 1$, $\lambda = 1$ and $p = 2$ in the loss functions.

In addition to the results in this section, we also provide more qualitative comparisons in the supplementary material and in the videos on the project website: `https://sites.google.com/a/umich.edu/rubenevillegas/iclr2017`.

Architectures. The content encoder of MCnet is built with the same architecture as VGG16 (Simonyan and Zisserman, 2015) up to the third pooling layer. The motion encoder of MCnet is also similar to VGG16 up to the third pooling layer, except that we replace its consecutive 3x3 convolutions with single 5x5, 5x5 and 7x7 convolutions in each layer. The combination layers are composed of 3 consecutive 3x3 convolutions (256, 128, and 256 channels in each layer). The multi-scale residuals are composed of 2 consecutive 3x3 convolutions. The decoder is the mirrored architecture as the content encoder where we perform unpooling followed by deconvolution. For the baseline ConvLSTM, we use the same architecture as the motion encoder, residual connections and decoder, except we increase the number of channels in the encoder in order to have a overall comparable number of parameters with MCnet.

6.1 KTH AND WEIZMANN ACTION DATASETS

Experimental settings. The KTH human action dataset (Schuldt et al., 2004) contains 6 categories of periodic motions on simple background: running, jogging, walking, boxing, hand-clapping and hand-waiving. We use the standard train/test split provided with the KTH dataset. We train our network and baseline by observing 10 frames and predicting 10 frames into the future on the KTH dataset. We set $\beta = 0.02$ for training. We also select the walking, running, one-hand waving, and two-hands waving sequences from the Weizmann action dataset (Gorelick et al., 2007) for testing the networks' generalizability.

For all the experiments we test the networks on predicting 20 time steps into the future. As evaluation, we use the same SSIM and PSNR metrics as in Mathieu et al. (2015). The evaluation on KTH was performed on sub-clips within each video in the testset. We sample sub-clips every 3 frames for running and jogging, and sample sub-clips every 20 frames (skipping the frames we have already predicted) for walking, boxing, hand-clapping and hand-waving. Sub-clips for running, jogging and walking were manually trimmed to ensure humans are always present in the frames. The evaluation on Weizmann was performed on all sub-clips in the selected sequences.

Results. Figure 2 summarizes the quantitative comparisons among our MCnet, ConvLSTM baseline and their residual variations. In the KTH test set, our network outperforms the ConvLSTM baseline by a small margin. However, when we test the residual versions of MCnet and ConvLSTM on the dataset (Gorelick et al., 2007) with similar motions, we can see that our network can generalize

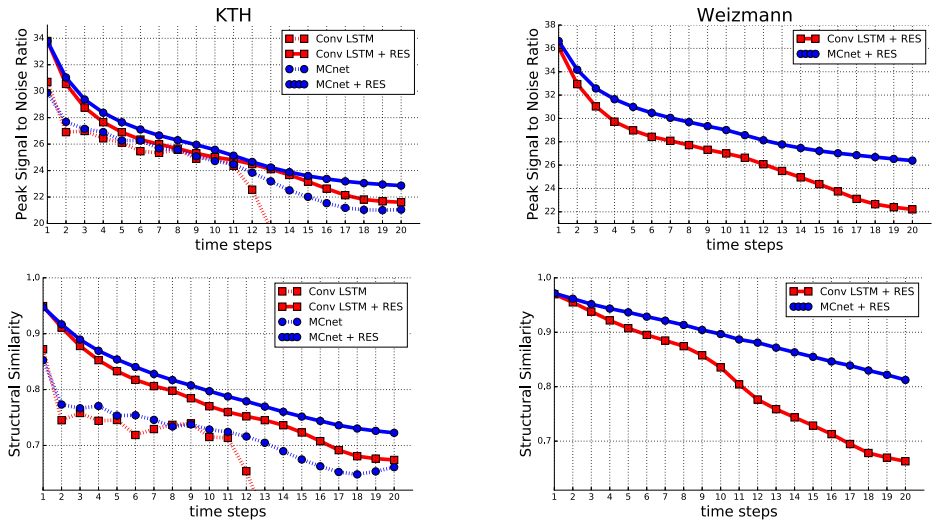


Figure 2: Quantitative comparison between MCnet and ConvLSTM baseline with and without multi-scale residual connections (indicated by "+ RES"). Given 10 input frames, the models predict 20 frames recursively, one by one. Left column: evaluation on KTH dataset (Schuldt et al., 2004). Right colum: evaluation on Weizmann (Gorelick et al., 2007) dataset.

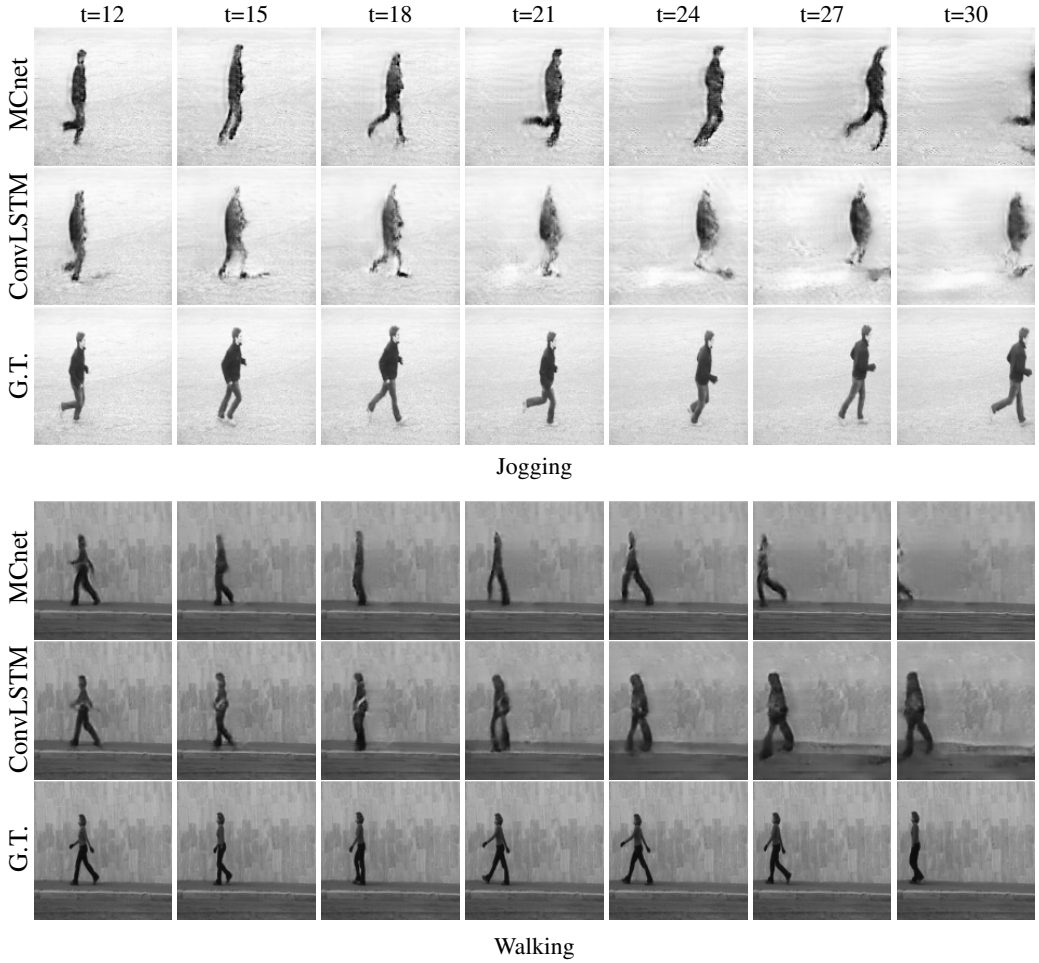


Figure 3: Qualitative comparison between our MCNet model and ConvLSTM. We display predictions starting from the 12^{th} frame, in every 3 timesteps. The first 3 rows correspond to KTH dataset for the action of jogging and the last 3 rows correspond to Weizmann dataset for the action of walking.

well to the unseen contents by showing clear improvements, especially in long-term prediction. One reason for this result is that the test and training partitions of the KTH dataset have simple and similar image contents so that ConvLSTM can memorize the average background and human appearance to make reasonable predictions. However, when tested on unseen data, ConvLSTM has to internally take care of both scene dynamics and image contents in a mingled representation, which gives it a hard time for generalization. In contract, the reason why our network outperforms the ConvLSTM baseline on unseen data is that our network focuses on identifying general motion features and applying them to a learned content representation.

Figure 3 presents qualitative results of multi-step prediction by our network and ConvLSTM. As expected, prediction results by our full architecture preserves human shapes more accurately than the baseline. It is worth noticing that our network produces very sharp prediction over long-term time steps; it shows that MCnet is able to capture periodic motion cycles, which reduces the uncertainty of future prediction significantly. More qualitative comparisons are shown in the supplementary material and the project website `https://goo.gl/nG8ve1`.

6.2 UCF-101 DATASET

Experimental settings. This section presents results on the challenging real-world videos in the UCF-101 (Soomro et al., 2012) dataset. Having collected from YouTube, the dataset contains 101 realistic human actions taken in a wild and exhibits various challenges, such as background clutter, occlusion, and complicated motion. We employed the same network architecture as in the KTH dataset, but trained the network to observe 4 frames and predict a single frame. We set $\beta = 0.001$ for training. We also trained our convolutional LSTM baseline in the same way. Following the same protocol as Mathieu et al. (2015) for data pre-processing and evaluation metrics on full images, all networks were trained on Sports-1M (Karpathy et al., 2014) dataset and tested on UCF-101 unless otherwise stated.[1]

Results. Figure 4 shows the quantitative comparisons between our network trained for single-step-prediction and Mathieu et al. (2015). We can clearly see the advantage of our network over the baseline. The separation of motion and contents in two encoder pathways allows our network to identify key motion and content features which are then fed into the decoder to yield predictions of higher quality compared to the baseline.[2] In other words, our network only moves what shows motion in the past, and leaves the rest untouched. We also trained a residual version of MCnet on UCF-101, indicated by "MCnet + RES UCF101", to compare how well our model generalizes when trained and tested on the same or different dataset(s). To our surprise, when tested on tested with UCF-101, the MCnet trained on Sports-1M (MCnet + RES) roughly matches the performance of the MCnet trained on UCF-101 (MCnet + RES UCF101), which suggests that our model learns effective representations which can generalize to new datasets. Figure 5 presents qualitative comparisons between frames generated by our network and Mathieu et al. (2015). Since the ConvLSTM and Mathieu et al. (2015) lack explicit motion and content modules, they lose sense of the dynamics in the video and therefore the contents become distorted quickly. More qualitative comparisons are shown in the supplementary material and the project website `https://goo.gl/nG8ve1`.

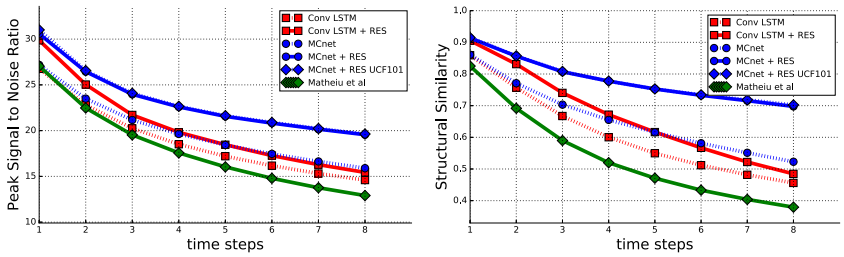


Figure 4: Quantitative comparison between our model, convolutional LSTM Shi et al. (2015), and Mathieu et al. (2015). Given 4 input frames, the models predict 8 frames recursively, one by one.

[1] We use the code and model released by Mathieu et al. (2015) at `https://github.com/coupriec/VideoPredictionICLR2016`

[2] We were not able to get the model fine-tuned on UCF-101 from the authors so it is not included in Figure 4

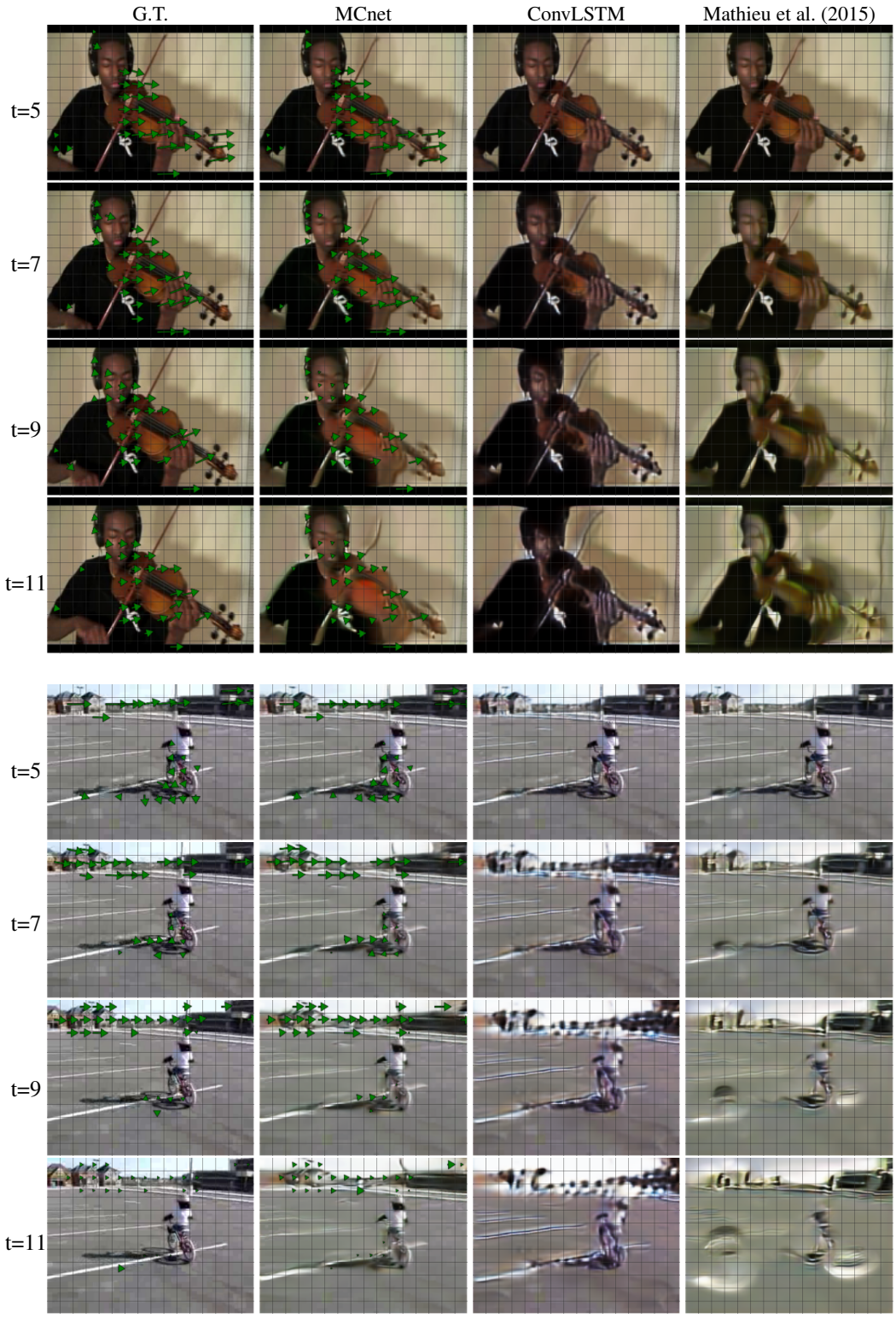


Figure 5: Qualitative comparisons among MCnet and ConvLSTM and Mathieu et al. (2015). We display predicted frames (in every other frame) starting from the 5^{th} frame. The green arrows denote the top-30 closest optical flow vectors within image patches between MCnet and ground-truth. More clear motion prediction can be seen in the videos at `https://goo.gl/nG8ve1`.

7 CONCLUSION

We proposed a motion-content network for pixel-level prediction of future frames in natural video sequences. The proposed model employs two separate encoding pathways, and learns to decompose motion and content without explicit constraints or separate training. Experimental results suggest that separate modeling of motion and content improves the quality of the pixel-level future prediction, and our model overall achieves state-of-the-art performance in predicting future frames in challenging real-world video datasets.

REFERENCES

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. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

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

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

M. Hoai and F. Torre. Max-margin early event detectors. *IJCV*, 2013.

A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

T. Lan, T. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*, 2014.

W. Lotter, G. Kreiman, and D. Cox. Unsupervised learning of visual structure using predictive generative networks. *arXiv preprint arXiv:1504.08023*, 2015.

M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

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

L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Scholkopf, and W. T. Freeman. Seeing the arrow of time. In *CVPR*, 2014.

S. L. Pintea, J. C. van Gemert, and A. W. M. Smeulders. Dejavu: Motion prediction in static images. In *European Conference on Computer Vision*, 2014.

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

M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, 2011.

C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004.

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 *Advances in Neural Information Processing Systems 28*. 2015.

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

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

K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

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

C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *arXiv preprint arXiv:1504.08023*, 2015.

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

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

J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. *CoRR*, abs/1606.07873, 2016. URL `http://arxiv.org/abs/1606.07873`.

P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *ICCV 2013 - IEEE International Conference on Computer Vision*, pages 1385–1392, Sydney, Australia, Dec. 2013. IEEE. doi: 10.1109/ICCV.2013.175. URL `https://hal.inria.fr/hal-00873592`.

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

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

M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2011.

8 APPENDIX

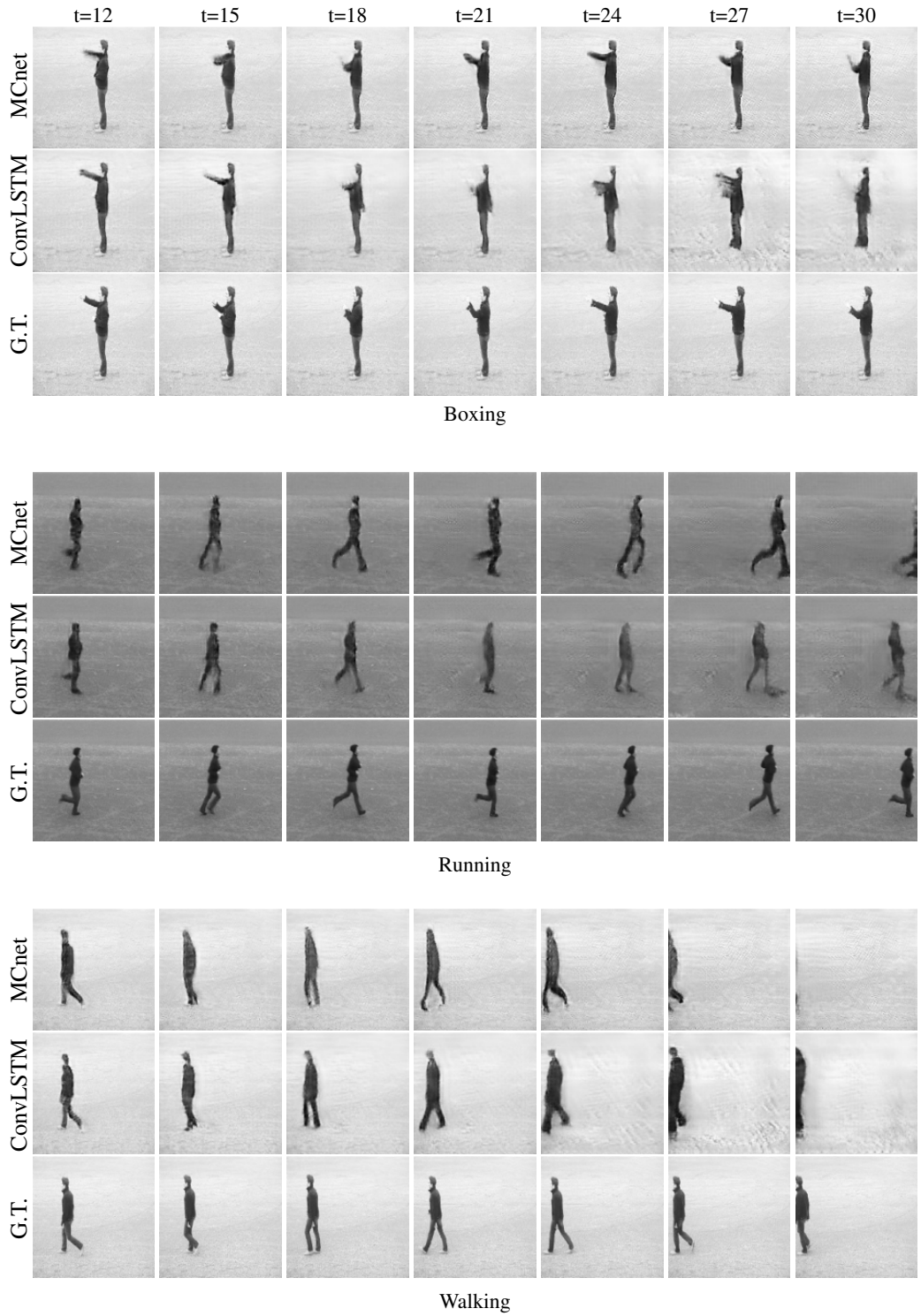


Figure 6: Qualitative comparisons on KTH testset. We display predictions starting from the 12^{th} frame, in every 3 timesteps.

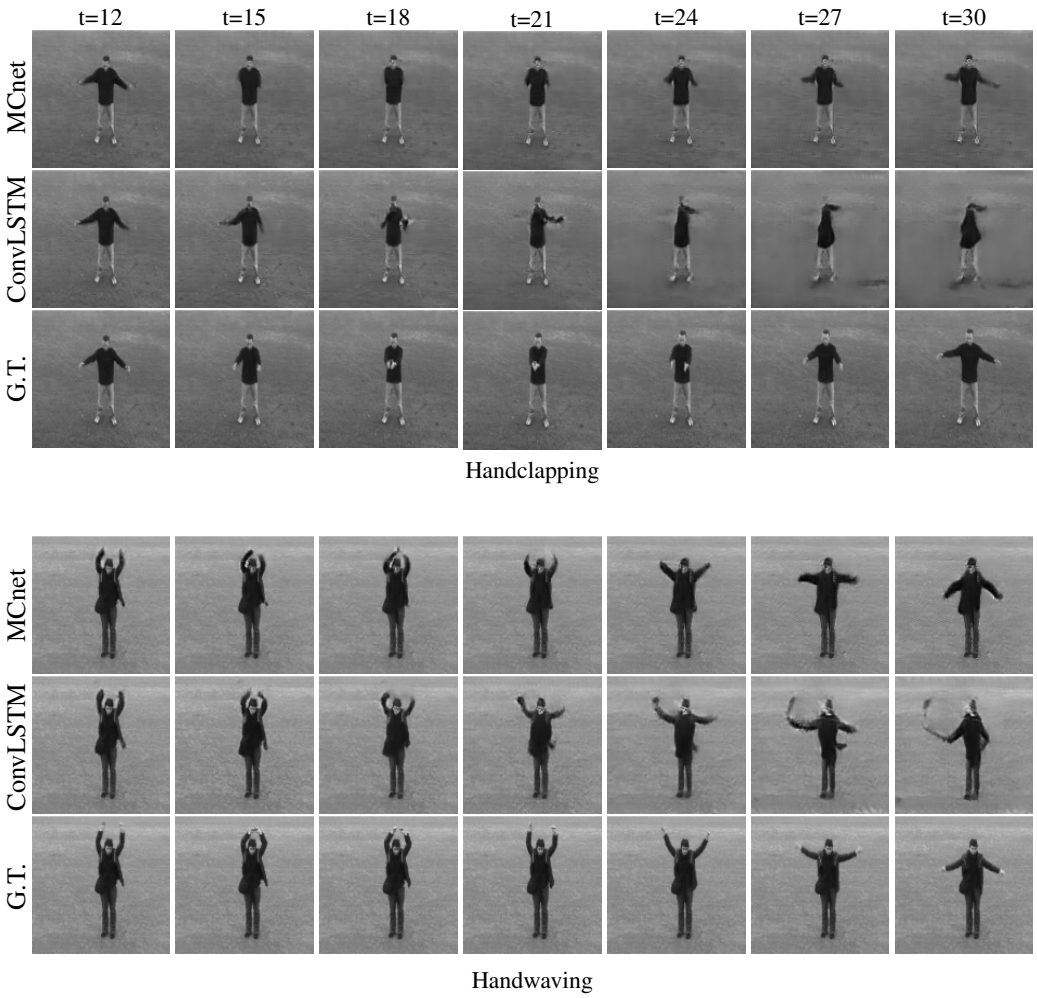


Figure 7: Qualitative comparisons on KTH testset. We display predictions starting from the 12^{th} frame, in every 3 timesteps.

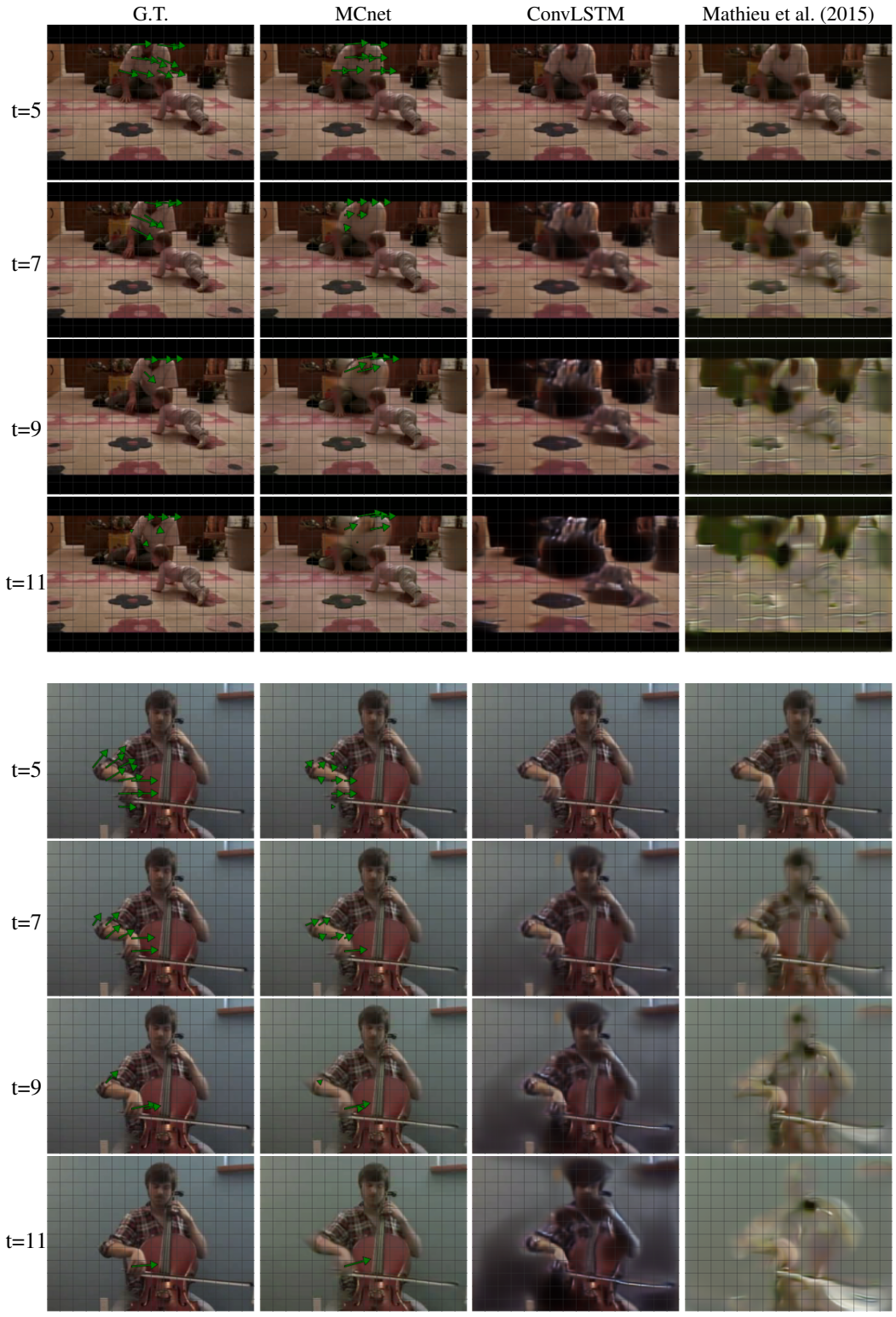


Figure 8: Qualitative comparisons on UCF-101. We display predictions (in every other frame) starting from the 5^{th} frame. The green arrows denote the top-30 closest optical flow vectors within image patches between MCnet and ground-truth. More clear motion prediction can be seen in the videos `https://goo.gl/nG8ve1`.

A QUALITATIVE AND QUANTITATIVE COMPARISON WITH CONSIDERABLE CAMERA MOTION AND ANALYSIS

In this section, we show frame prediction examples in which considerable camera motion occurs. We analyze the effects of camera motion on our best network and the corresponding baselines. First, we analyze qualitative examples on UCF101 (more complicated camera motion) and then on KTH (zoom-in and zoom-out camera effect).

UCF101 results. As we can see in Figure 9 and Figure 10, our model does a reasonable job at handling foreground and camera motion up to some point in the predicted sequence. We hypothesize that for the first few steps, motion signals from images are clear. However, as images are predicted, motion signals start to deteriorate due to prediction errors. When considerable camera motion is present in image sequences, the motion signals are very dense. As predictions evolve into the future, our motion encoder has to handle large motion deterioration due to prediction errors which cause motion signals to get easily confused and lost quickly.

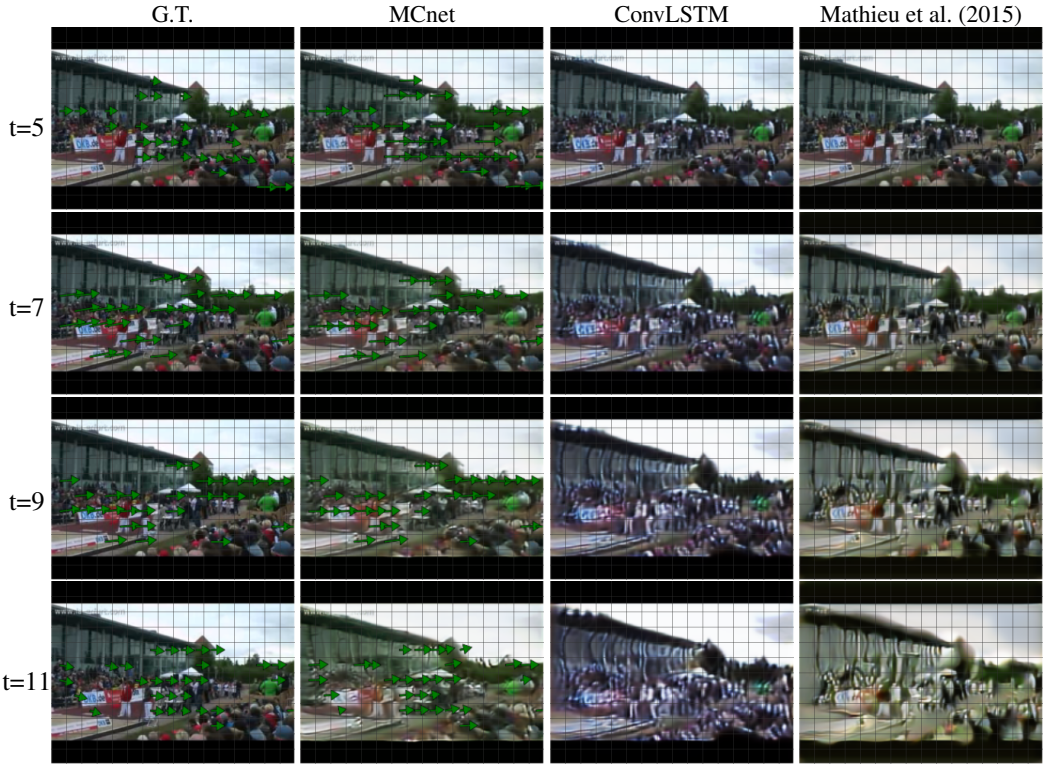


Figure 9: Qualitative comparisons on UCF-101. We display predictions (in every other frame) starting from the 5^{th} frame. The green arrows denote the top-30 closest optical flow vectors within image patches between MCnet and ground-truth. More clear motion prediction can be seen in the videos `https://goo.gl/nG8ve1`.

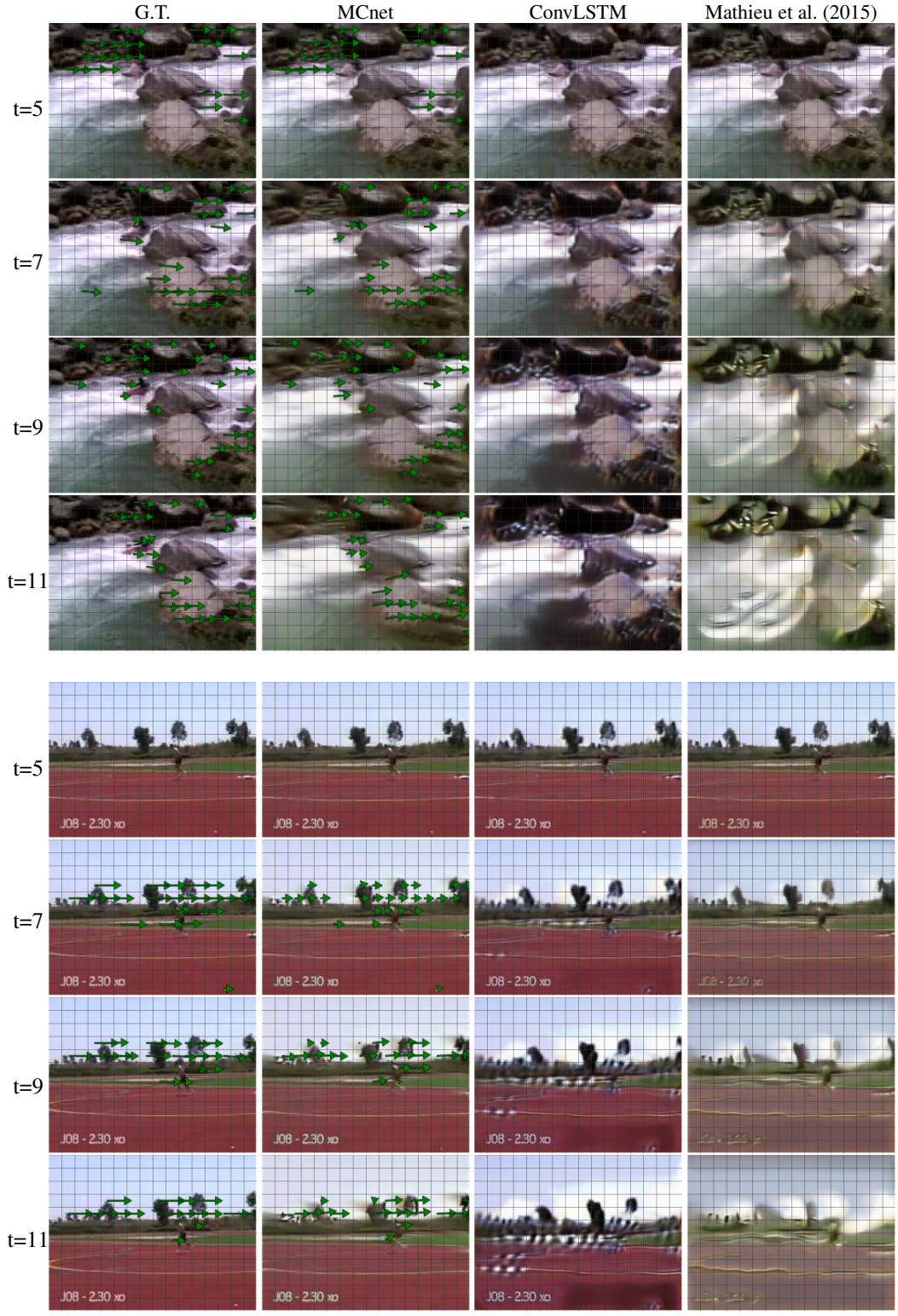


Figure 10: Qualitative comparisons on UCF-101. We display predictions (in every other frame) starting from the 5^{th} frame. The green arrows denote the top-30 closest optical flow vectors within image patches between MCnet and ground-truth. More clear motion prediction can be seen in the videos `https://goo.gl/nG8ve1`.

KTH results. We were unable to find videos with background motion in the KTH dataset, but we found videos where the camera is zooming-in or out in the actions of boxing, handclapping and handwaving. In Figure 11 and Figure 12, we display qualitative for such videos. Our model is able to predict the zoom change in the cameras reasonably well while continuing the action motion. In comparison to the performance observed in UCF101, the background does not change much. Thus, the motion signals are well localized in the foreground motion (human), and do not get confused with the background and lost as quickly.

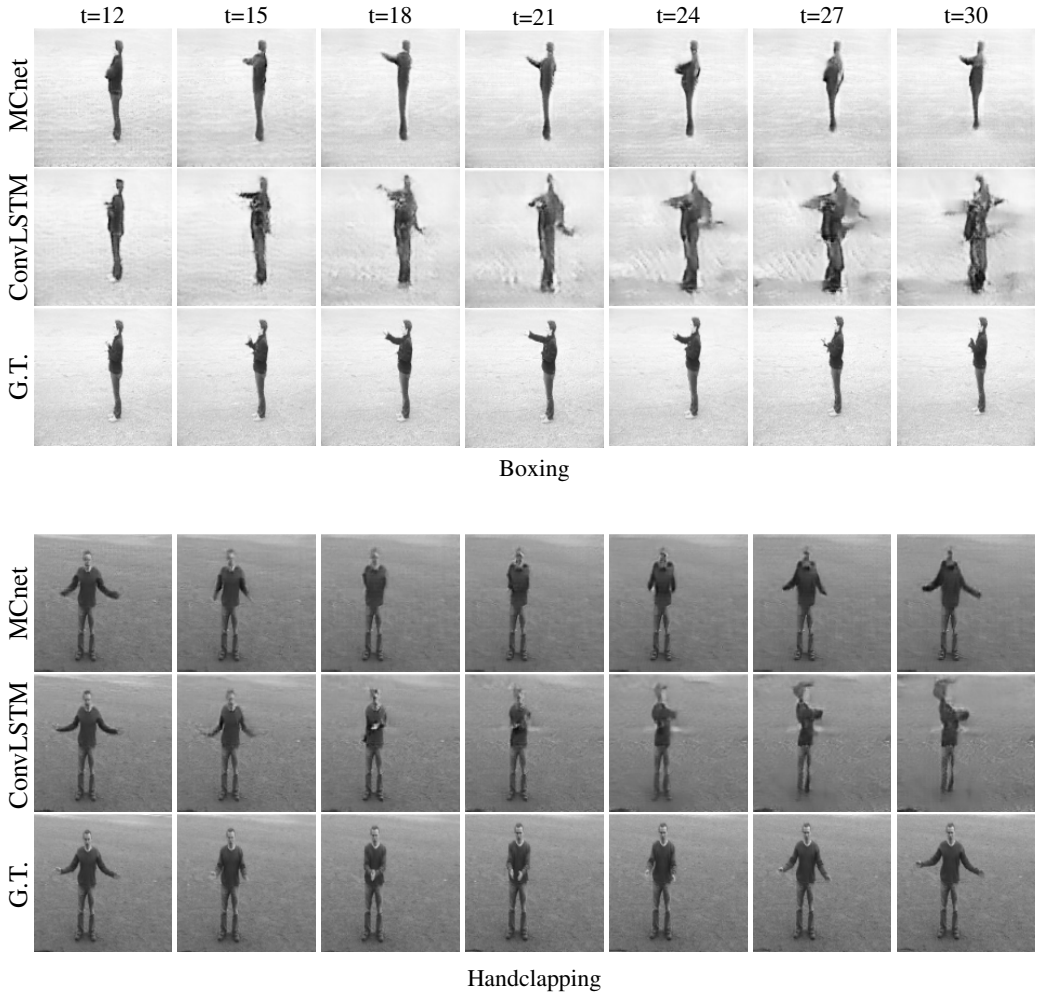


Figure 11: Qualitative comparisons on KTH testset. We display predictions starting from the 12^{th} frame, in every 3 timesteps. More clear motion prediction can be seen in the videos `https://goo.gl/nG8ve1`.

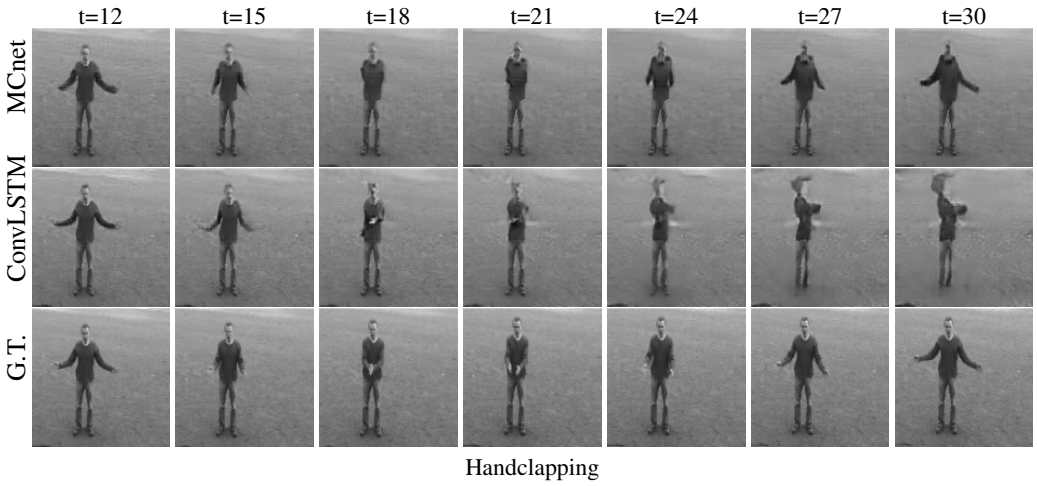


Figure 12: Qualitative comparisons on KTH testset. We display predictions starting from the 12^{th} frame, in every 3 timesteps. More clear motion prediction can be seen in the videos `https://goo.gl/nG8ve1`.

B EXTENDED QUANTITATIVE EVALUATION

In this section, we show additional quantitative comparison with a baseline based on copying the last observed frame through time for KTH and UCF101 datasets. Copying the last observed frame through time ensures perfect background prediction in videos where most of the motion comes from foreground (i.e. person performing an action). However, if such foreground composes a small part of the video, it will result in high prediction quality score regardless of the simple copying action.

In Figure 13, we can see the quantitative comparison in the datasets. Copying the last observed frame through time does a reasonable job in both datasets, however, the impact is larger in UCF101. Videos in the KTH dataset comprise simple backgrounds with minimal camera motion, which allows our network to easily predict both foreground and background motion, resulting in better image quality scores. However, videos in UCF101 contain more complicated and diverse background which in combination with camera motion present a much greater challenge to video prediction networks. From the qualitative results in Section A and Figures 5, 8, 9, and 10, we can see that our network performs better in videos that contain isolated areas of motion compared to videos with dense motion. A simple copy paste operation of the last observed frame, ensures very high prediction scores in videos where very small motion occurs. The considerable score boost by videos with small motion causes the simple copy paste baseline to outperform MCnet in the overall performance on UCF101.

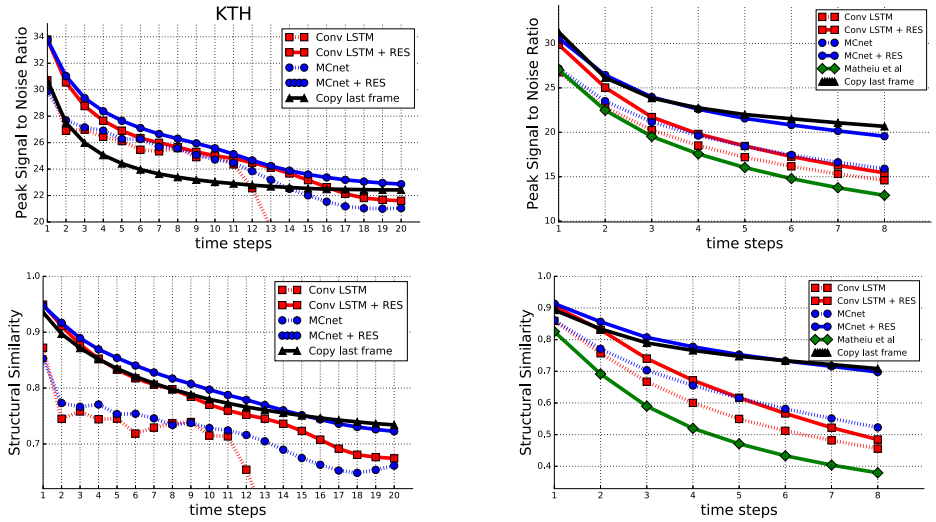


Figure 13: Extended quantitative comparison including a baseline based on copying the last observed frame through time.

C UCF101 MOTION DISAMBIGUATION EXPERIMENTS

Due to the observed bias from videos with small motion, we perform experiments by measuring the image quality scores on areas of motion. These experiments are similar to the ones performed in Mathieu et al. (2015). We compute DeepFlow optical flow (Weinzaepfel et al., 2013) between the previous and the current groundtruth image of interest, compute the magnitude, and normalize it to $[0, 1]$. The computed optical flow magnitude is used to mask the pixels where motion was observed. We set the pixels where the optical flow magnitude is less than 0.2 to 0, and leave everything else untouched in both the groundtruth and predicted images. Additionally, we separate the test videos by the average ℓ_2-norm of time difference between target frames. We separate the test videos into deciles based of the computed average ℓ_2-norms, and compute image quality on each decile. Intuitively, the 1^{st} decile contains videos with the least overall of motion (i.e. frames show smallest change over time), and the 10^{th} decile contains videos with the most overall motion (i.e. frames show largest change over time).

In Figure 14, we see that when we only evaluate on pixels where rough motion is observed, MCnet reflects higher PSNR and SSIM, and clearly outperforms all the baselines in terms of SSIM. The SSIM results show that our network is able to predict a structure (i.e. textures, edges, etc) similar to the grountruth images within the areas of motion. The PSNR results, however, show that our method outperforms the simple copy paste baseline for the first few steps but then our method performs slightly worse. The discrepancies observed between PSNR and SSIM scores could be due to the fact that some of the predicted images may not reflect the exact pixel values of the groundtruth regardless of the structures being similar. SSIM scores are known to take into consideration features in the image that go beyond directly matching pixel values, reflecting more accurately how humans perceived image quality.

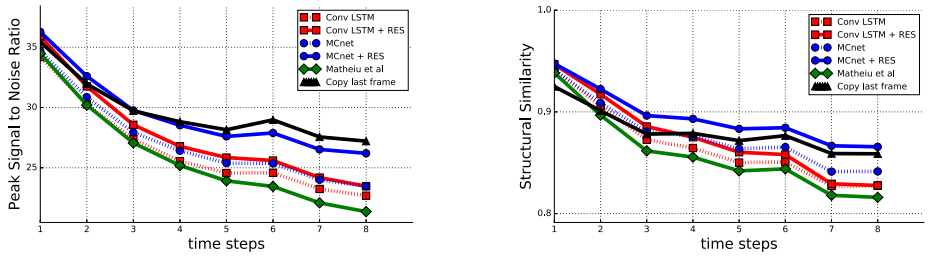


Figure 14: Extended quantitative comparison on UCF101 including a baseline based on copying the last observed frame through time using motion based pixel mask.

In Figures 16 and 15, we show evaluation by separating the test videos into deciles based on the average ℓ_2-norm of time difference between target frames. From this evaluation, we can see that the baseline based on copying the last frame scores higher in videos where motion is the smallest. The first few deciles (videos with small motion) show that, our network is not just copying the last observed frame through time, otherwise it would perform similarly to the copy last frame baseline. The last deciles (videos with large motion) show our network outperforming all the baselines, including the copy previous baseline, effectively confirming that our network does predict motion similar to the motion observed in the video.

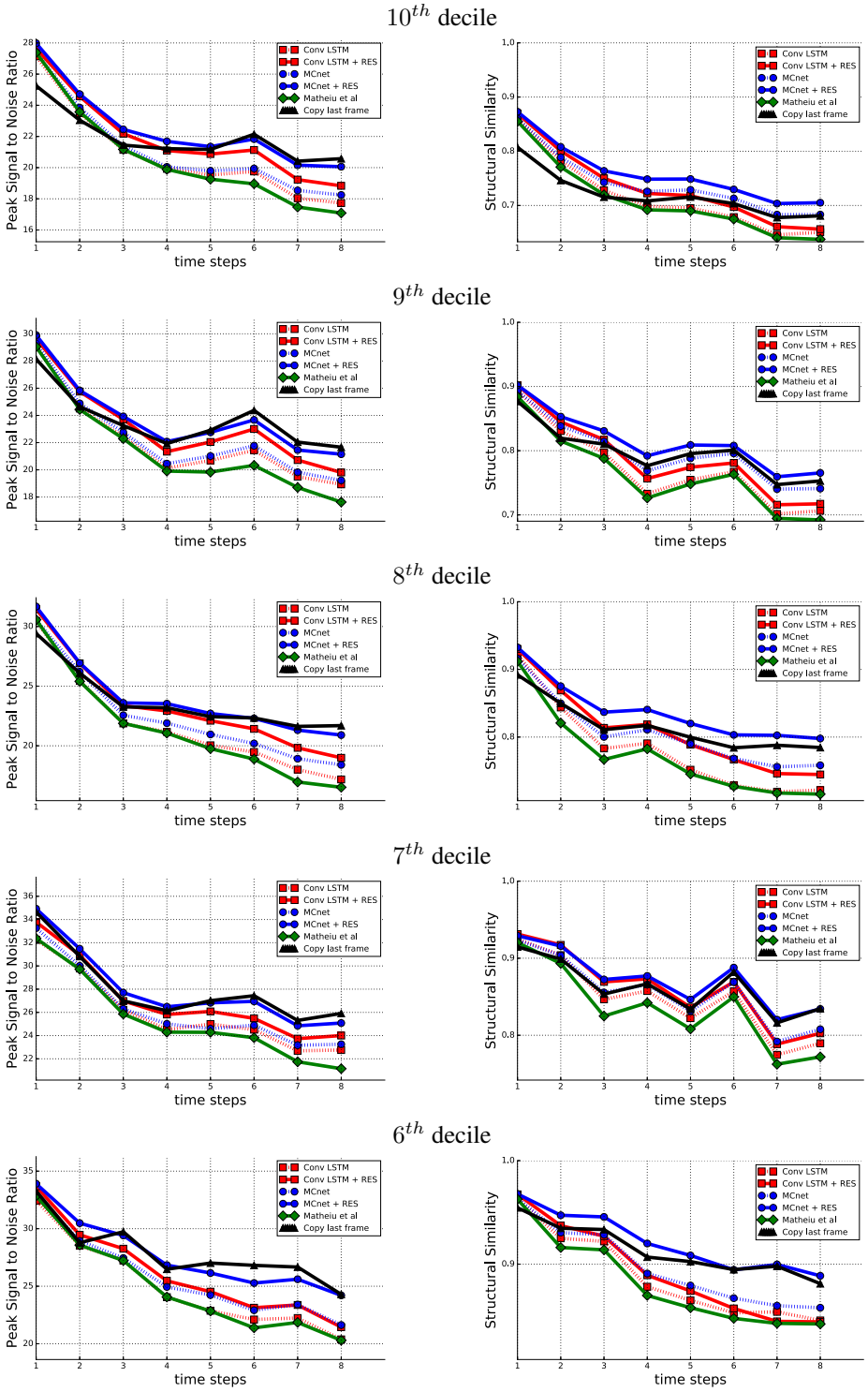


Figure 15: Quantitative comparison on UCF101 using motion based pixel mask, and separating dataset by average ℓ_2-norm of time difference between target frames.

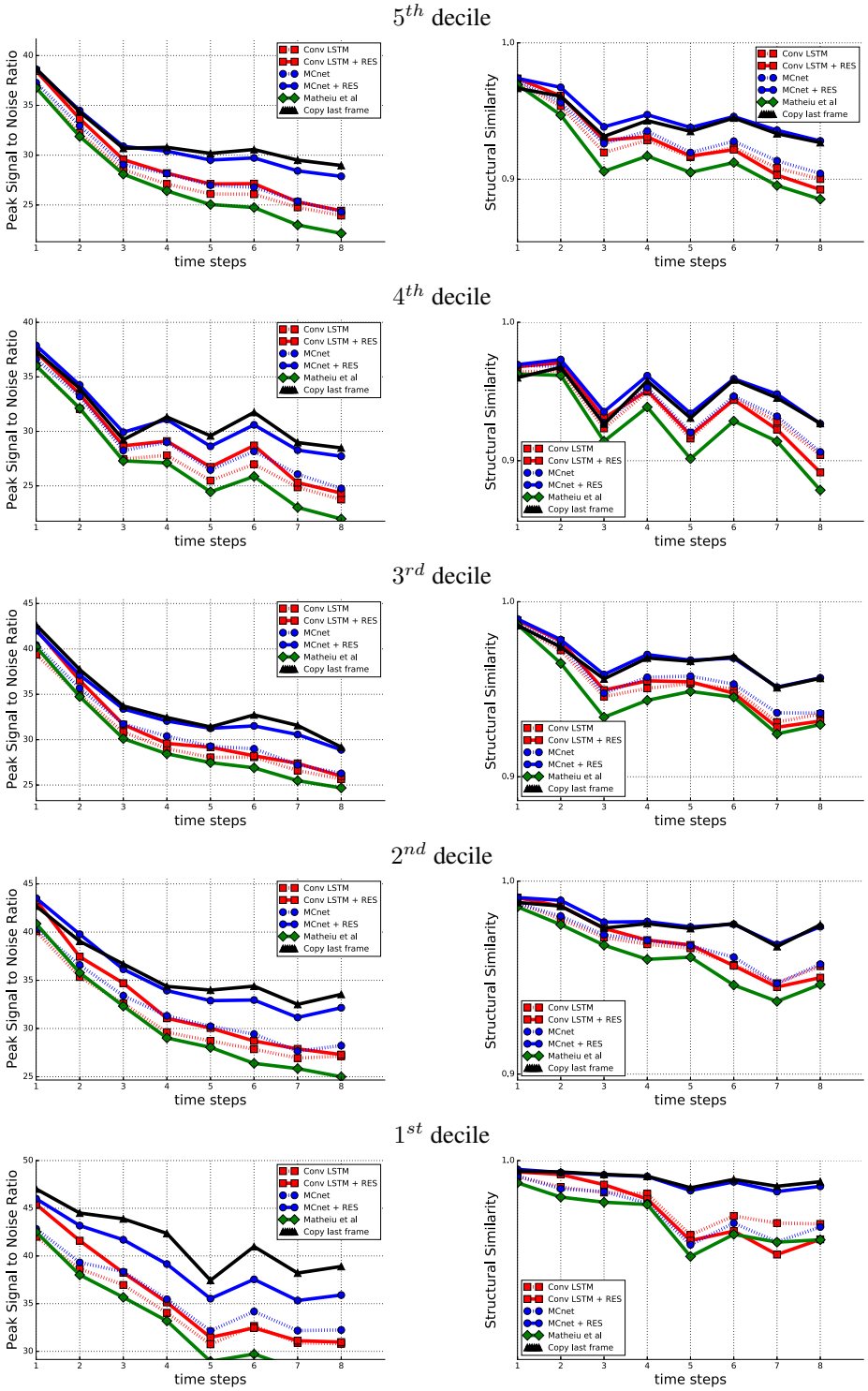


Figure 16: Quantitative comparison on UCF101 using motion based pixel mask, and separating dataset by average ℓ_2-norm of time difference between target frames.

D ADVERSARIAL TRAINING

Mathieu et al. (2015) proposed an adversarial training for frame prediction. Inspired by Goodfellow et al. (2014), they propose a training procedure that involves a generative model G and a discriminative

model D. The two models compete in a two-player minimax game. The discriminator D is optimized to correctly classify its inputs as either coming from the training data (real frame sequence) or from the generator G (synthetic frame sequence). The generator G is optimized to generate frames that *fool* the discriminator into believing that they come from the training data. At training time, D takes the concatenation of the input frames that go into G and the images produced by G. The adversarial training objective is defined as follows

$$\min_G \max_D \ \log D\left([\mathbf{x}_{1:t}, \mathbf{x}_{t+1:t+T}]\right) + \log\left(1 - D\left([\mathbf{x}_{1:t}, G\left(\mathbf{x}_{1:t}\right)]\right)\right),$$

where $[.,.]$ denotes concatenation in the depth dimension, $\mathbf{x}_{1:t}$ denote the input frames to G, $\mathbf{x}_{t+1:t+T}$ are the target frames, and $G\left(\mathbf{x}_{1:t}\right) = \hat{\mathbf{x}}_{t+1:t+T}$ are the frames predicted by G. In practice, we split the minimax objective into two separate, but equivalent, objectives: $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{disc}}$. During optimization, we minimize the adversarial objective alternating between $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{disc}}$. $\mathcal{L}_{\text{GAN}}$ is defined by

$$\mathcal{L}_{\text{GAN}} = -\log D\left([\mathbf{x}_{1:t}, G\left(\mathbf{x}_{1:t}\right)]\right),$$

where we optimize the parameters of G to minimize $\mathcal{L}_{\text{GAN}}$ while the parameters of D stay untouched. As a result, G is optimized to generate images that make D believe that they come from the training data. Thus, the generated images look sharper, and more realistic. $\mathcal{L}_{\text{disc}}$ is defined by

$$\mathcal{L}_{\text{disc}} = -\log D\left([\mathbf{x}_{1:t}, \mathbf{x}_{t+1:t+T}]\right) - \log\left(1 - D\left([\mathbf{x}_{1:t}, G\left(\mathbf{x}_{1:t}\right)]\right)\right),$$

where we optimize the parameters of D to minimize $\mathcal{L}_{\text{disc}}$ while the parameters of G stay untouched. D learns to tell whether its input came from the training data or the generator G. Alternating between the two objectives, causes G to generate very realistic images and D to not be able to distinguish between generated sequences of frames and sequences of frames from the training data.