CDNET: A CASCADED DECOUPLING ARCHITECTURE FOR VIDEO PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video prediction is an essential task in the computer vision community, helping to solve many downstream vision tasks by predicting and modeling future motion dynamics and appearance. In the deterministic video prediction task, current methods mainly employ variants of stacked Recurrent Neural Networks (RNN) to capture spatiotemporal coherence, overlooking the conflict between long-term motion dynamics modeling and legible appearance generation. In this work, we propose a Cascaded Decoupling Network (CDNet) to solve the video prediction problem through two modules: motion LSTM to capture the motion trend and variation in the temporal highway without considering the appearance details, and refine LSTM to recover the detailed appearance according to the predicted motion dynamics and historical appearance iteratively. The cascaded structure provides a preliminary solution for the above conflict. We verify the rationality of our model on two real-world challenging video prediction datasets and yield state-of-the-art performance.

1 INTRODUCTION

Video prediction is the task of predicting future video frames conditioned on a few observed video frames. Recently, it has attracted increasing attention for its self-supervised spatiotemporal feature extraction ability, which benefits downstream visual tasks, such as Video Question Answer (Jing et al., 2020), rainfall forecasting (Shi et al., 2015), robot motion planning (Finn et al., 2016), and autonomous driving (Kwon & Park, 2019a). In our work, we focus on the deterministic self-supervised video prediction task, which needs no intermediate information, such as semantic labels (Lee et al., 2021), optical flow (Wu et al., 2020), or pre-trained model from other tasks, to guide the frame synthesis. In this task, the modeling of motion dynamics and appearance is crucial for plausible video prediction.

Recent state-of-the-art deterministic video prediction approaches focus on capturing spatiotemporal coherence by stacked LSTM variants architectures (Wang et al., 2017; 2018a; 2019), motion dynamics analyses architectures (Jin et al., 2020; Wu et al., 2021), or 3D convolutional architectures (Wang et al., 2018b; Yu et al., 2020). In these approaches, the modeling of complex motion dynamics and the generation of detailed frames are simultaneously done in one LSTM cell, so that the long-term trend forecasting and specific pixel value prediction, which are often in conflict with each other, need to be balanced at each time step. Over time, the simple LSTM will gradually forget the historical appearance information, leading to significant degradation of the future appearance prediction.

Motivated by the above observations, we propose to use explicit structures to decouple motion dynamics and appearance information to reduce conflicts in predictions. Referring to the habits of humans, we hypothesize that the recovery and refinement of the appearance information depend on the prediction of motion dynamics, and design a cascade network to decouple the motion dynamics and appearance information. Previous decoupling methods (Villegas et al., 2017; Guen & Thome, 2020) used the residual structure to predict the appearance which still models motion information, ignoring the above-mentioned conflicts. While our network divides motion prediction and appearance prediction into two individual processes in an end-to-end architecture. Firstly, our network predicts motion dynamics for the next time step and generates a new position feature. Then, according to the updated motion state, it synthesizes the new appearance of the foreground and recovers the vacancy of the background. Such a cascaded decoupling method can not only make a plausible prediction for complex motion but also generate a legible frame according to the deterministic motion results.

However, it is difficult to keep observed long-term appearance information by simple cascaded architecture. As in the Recurrent Neural Network, the forget gate usually forgets certain information which is insignificant for current frame prediction or appearance information which is difficult to predict in the future. To solve this drawback, a direct way is using self-attention (Bahdanau et al., 2015) at every time step (Lin et al., 2020) which requires a huge number of parameters to store keys and values for each frame. In our architecture, inspired by the structure of sequence to sequence information transfer, we extract global foreground and background appearance information and store them in external memory, and design a novel refine LSTM cell to unite the global appearance information and current appearance information iteratively. Thus, our network could refine the appearance prediction at each refinement iterations by referring to the observed global appearance information. Specifically, we reuse a single cell for the refining process, which is elastic in the refining process for different data and reduces the number of parameters compared with the current prevailing stacked LSTM architecture (Wang et al., 2017; Yu et al., 2020).

In order to clearly decouple the motion dynamics estimation and the appearance refinement process, we assign the two tasks to two different structures with different losses. Specifically, a powerful LSTM-based model is employed to capture the motion dynamics. Then, based on the current motion dynamics, a cyclic refinement module with information integration capabilities is employed for the filling and refinement of pixels. There is no temporal information transfer between the refinement modules of the same layer at adjacent moments. We use the hierarchical constraint for different processes in the pixel refinement phase and motion area constraint for motion dynamics prediction. The pixel loss gives decreasing restriction from the final refinement iteration to the beginning refinement iteration to enhance the refining ability in appearance prediction, while the area loss provides a looser constraint for motion dynamics prediction which convergences to predict the motion area of objects. Based on the above structures and losses, we divide the difficult pixel-level prediction task into motion dynamics modeling and appearance recovery. Compared with normal stacked RNN architecture, our architecture can deal with gradient diffusion in the deep architecture, since our loss can directly pass down to the beginning process of the refining model and coarse motion prediction layer. The contributions of our work can be summarized as:

1) We propose a Cascaded Decoupling Network (CDNet) together with two loss functions to explicitly decouple motion dynamics and appearance information for video prediction.

2) We design a novel refinement LSTM unit, which can integrate predicted frame feature and global appearance information iteratively to refine the prediction results.

3) The proposed architecture achieves state-of-the-art performances in two real-world video datasets.

2 Related Work

According to the forecasting reference and forecasting ways, most video prediction methods can be classified as direct pixel prediction (Wang et al., 2017; Guen & Thome, 2020), explicit transform prediction (Reda et al., 2018), and trend probabilistic prediction (Chiu et al., 2020; Kwon & Park, 2019b). In our work, we focus on the self-supervised direct pixel prediction task.

RNN based methods have recently achieved promising results in video prediction. (Shi et al., 2015) initially proposed to replace the fully connected network with the convolution network in RNN gating control to extract spatiotemporal information, providing a powerful base model for subsequent networks. To enhance the spatiotemporal coherence, (Wang et al., 2017) proposed a horizontal memory flowing in the zigzag direction. Considering the linear translation restriction of gates in Long-Short Temporal Memory, (Wang et al., 2018a) and (Wang et al., 2019) introduced cascaded gate structures for forgetting gate and output gate, respectively. (Wang et al., 2018b; Yu et al., 2020) proved that 3D convolution architecture is useful for video prediction task by capturing the local temporal relation. (Lin et al., 2020) introduced a global attention module embedded in the traditional convolutional LSTM that the prediction of each time step depends on the correlation between current features and past features.

Considering the complex variations within the motions, (Guen & Thome, 2020) and (Wu et al., 2021) proposed specific motion dynamic capture units for motions in videos. (Guen & Thome, 2020) disentangled the motion dynamics into known physical dynamics and unknown factors. The motion unit leveraged partial differential equations to capture the motion dynamics. (Wu et al., 2021) presented a MotionGRU unit to inference the transient variation and the motion trend simultaneously.

We also adopt decoupling architecture for video prediction. Different from previous decoupling methods (Guen & Thome, 2020), our architecture explicitly decouples the motion dynamics prediction and appearance prediction in a cascade structure. The refinement module could iteratively refine the details in both the foreground and background based on the motion prediction results. For preserving global information, our architecture integrates the global appearance information in a single memory, which reduces the correlation search space from past time dimension to one.

3 Approach

Video prediction is to extrapolate future video frames based on the observed video frames. To unify the symbolic identification, we define the observed frame and predicted frame as x and $\hat{x}$, respectively. Given the observed frames $x_{1:t} = \{x_1, x_2, ..., x_t\}$, we predict the future frames $\hat{x}_{t+1:t+T} = \{\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+T}\}$ for T time steps. In our work, we iteratively generate new frames based on the previous frames by using recurrent neural network.

Fig. 1 illustrates the pipeline of stacked ConvLSTM (Xingjian et al., 2015) and our Cascaded Decoupling Network (CDNet). Note that the stacked ConvLSTM directly predicts spatial-temporal evolution with multi-layers for complex structures in videos. While our CDNet predicts the motion and appearance in a cascade manner. Inspired by the observation that when facing an ambiguous future, human beings tend to obtain a plausible prediction of motion trends and a legible frame, we decouple the video prediction task into the temporal motion dynamics prediction module and the spatial appearance refinement module. Compared with previous residual decoupling methods (Guen & Thome, 2020; Villegas et al., 2017), we cascade these two modules to synthesize convincing frames. The refinement module relies on the output of the motion prediction module in an end-to-end architecture.

As diagrammed in Fig. 1(b), the prediction of frames is achieved by frame encoding, motion dynamics prediction, appearance refinement, and frame decoding. To preserve the observed foreground and background information, a global information integration process is added during the generation of future frames.

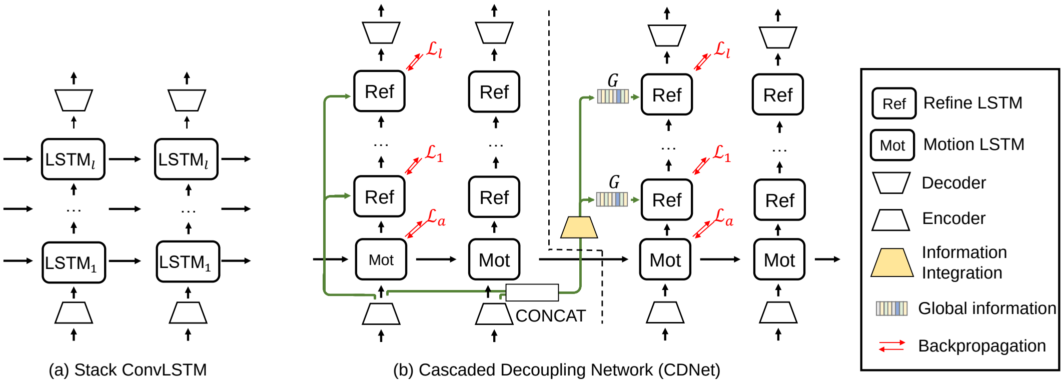
3.1 CDNet

Encoder and Decoder. The frame encoder extracts the spatial feature of the current input, while the decoder recovers the frames from the predicted next frame feature. In this work, we use the reversible encoder-decoder module with 3D convolution kernel (Yu et al., 2020) which extracts the feature by a two-way crossed encoder g^{Enc}, represented by $F_k^{enc} = g^{Enc}(x_{k:k+2})$ at time k.

Motion LSTM. As shown in Fig. 1(b), The bottom of our CDNet is the motion LSTM, which is the only explicit connection in the temporal dimension in the CDNet. Given observed frames, the CDNet first extrapolates the motion dynamics for individual objects in the current frame by one-layer LSTM and ignores the pixel changing. Two constraints are employed to decouple the motion dynamics from the input. In structure, future motion dynamics prediction relies on the temporal coherent capturing of this layer. In loss design, we abandon fine-grained pixel error and adopt motion change constraint, which will be elaborated in Section 3.3.

The motion LSTM is implemented by the ConvLSTM with memory cell (Wang et al., 2017). The input of the motion LSTM cell includes the encoding feature F_{k-1}^{enc}, the spatial information H_{k-1}, the temporal dynamics C_{k-1}, and spatiotemporal coherence M_{k-1} at previous time $k-1$. The output is the motion changing feature F_{k-1}^{dyn}

$$\left[F_k^{dyn}, H_k, C_k, M_k\right] = f^{Mot}\left(F_k^{enc}, H_{k-1}, C_{k-1}, M_{k-1}\right), \tag{1}$$



(a) Stack ConvLSTM

(b) Cascaded Decoupling Network (CDNet)

Figure 1: Pipeline of our framework: (a) Stack ConvLSTM (Xingjian et al., 2015). (b) Our CDNet. The CDNet consists of four modeling components: Encoder-Decoder module, motion LSTM for motion dynamics prediction, global information integration for preserving global appearance information, and refine LSTM for appearance refinement.

where f^{Mot} denotes the motion LSTM module. The $[H_k, C_k]$ is passed directly to the next moment for motion modeling. The $\left[F_k^{dyn}, M_k\right]$ is passed into the refine LSTM to provide intermediate features after motion prediction.

Refine LSTM. The refine LSTM is designed for appearance recovery and can refine the predicted frame from coarse-grained to fine-grained. It predicts a new prediction feature iteratively by reusing one cell, which also reduces the number of parameters compared with stacked ConvLSTM. With the motion dynamics changed but appearance preserved input F_k^{dyn}, the Refine LSTM generates the new prediction feature at layer l by

$$\left[F_k^{ref}, H_k^l, C_k^l, M_k^l\right] = f^{Ref}\left(F_k^{ref^{l-1}}, H_k^{l-1}, C_k^{l-1}, M_k^{l-1}, F_k^{enc}\right), \qquad (2)$$

where f^{Ref} denotes the refine LSTM module. We show the detail of the refine LSTM cell in Section 3.2. At the beginning phase of the refine LSTM, the refine prediction $F_k^{ref^{l-1}}$ and the spatiotemporal coherence M_k^{l-1} are equal with the outputs of the motion LSTM: $F_k^{ref^{l-1}} = F_k^{dyn}, M_k^{l-1} = M_k$. Since pixel information is omitted during the motion LSTM prediction process, we re-add pixel information F_k^{enc} during refinement.

Global information integration. During the prediction phase for $\hat{x}_{t+1:t+T}$, the LSTM easily forgets the long-term spatial information. In our work, we propose a global information preserving memory in a seq2seq way to memorize the past appearance in both foreground and background. The global information G is extracted by a multi-layer CNN structure,

$$\mathrm{G} = \{\mathrm{Relu}\left(\mathcal{W} * F_{1:t}^{enc} + b\right)\}_{\mathrm{p}}. \qquad (3)$$

where $*$ is convolution operator with corresponding weights $\mathcal{W}$ and bias b, and the p is the layer number of the convolution. Given the global information, the appearance preserving input F_k^{dyn} is replaced by G.

3.2 Refine LSTM cell

We design the refine LSTM cell to refine the predicted frame from coarse-grained to fine-grained. In the refine LSTM module, temporal motion dynamics transformation is cut from two adjacent moments, which makes the refine LSTM cell focus on the appearance refining and the recovery of motion area. As formulated in Equation 2, the feature prediction simultaneously refers to the output of the previous LSTM module and global appearance information. For the prediction of refinement feature, the refine LSTM first generates the new appearance update state C^l and new spatiotemporal

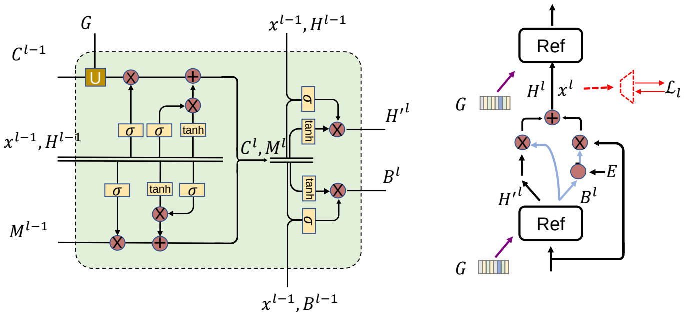


Figure 2: (Left) The design for the refine LSTM cell. The refine LSTM cell iteratively refines the predicted feature referring to the global foreground and background information. (Right) The hidden state update strategy. The hidden state H refers to the mask B to update the information, i.e. shielding the unchanged part and updating the changed part.

coherence M^l at level l as follows:

$$\begin{aligned}
g_c &= \tanh\left(\mathcal{W}_g * \left[x^{l-1}; H^{l-1}\right] + b_g\right)\\
i_c &= \sigma\left(\mathcal{W}_i * \left[x^{l-1}; H^{l-1}\right] + b_i\right)\\
f_c &= \sigma\left(\mathcal{W}_f * \left[x^{l-1}; H^{l-1}\right] + b_f\right)\\
C^l &= f_c \odot U\left(C^{l-1}, G\right) + i_c \odot g_c\\
g_m &= \tanh\left(\mathcal{W}'_g * \left[x^{l-1}; M^{l-1}\right] + b'_g\right)\\
i_m &= \sigma\left(\mathcal{W}'_i * \left[x^{l-1}; M^{l-1}\right] + b'_i\right)\\
f_m &= \sigma\left(\mathcal{W}'_f * \left[x^{l-1}; M^{l-1}\right] + b'_f\right)\\
M^l &= f_m \odot M^{l-1} + i_m \odot g_m,
\end{aligned} \tag{4}$$

where $\odot$ indicates Hadamard product. The cell gate activation function uses sigmoid function σ. The g, i, f, o are gates of LSTM which control the information flow from coarse-grained to fine-grained. The updater U is a multi-layer CNN structure to reintegrate the historical appearance information G into the appearance update state C^{l-1}. At each iteration, the appearance update state C refreshes its state according to its new refinement state. According to the new appearance update state C^l and new spatiotemporal coherence M^l, we update predicted frame feature H^l and information preserving mask state B^l, referring their previous state:

$$\begin{aligned}
o_h &= \sigma\left(\mathcal{W}_o * \left[x^{l-1}; H^{l-1}; C^l; M^l\right] + b_o\right)\\
\hat{H}^l &= o_h \odot \tanh\left(\mathcal{W}_{1\times 1} * \left[C^l, M^l\right]\right)\\
o_k &= \sigma\left(\mathcal{W}_k * \left[x^{l-1}; B^{l-1}; C^l; M^l\right] + b_k\right)\\
B^l &= o_k \odot \tanh\left(\mathcal{W}_{1\times 1} * \left[C^l, M^l\right]\right).
\end{aligned} \tag{5}$$

As shown in Fig. 3.2 (right part), to explicitly preserve previous unchanging features in the background and foreground during the iteratively refining process, we generate the predicted frame feature $\hat{H}^l$ and mask stateB^l at the same time. Then, the updated hidden state of the l_{th} layer can be calculated by:

$$H^l = \hat{H}^l \odot B^l \ + H^{l-1} \odot (E - B^l), \tag{6}$$

where E is a matrix with single value $\{1\}$. In this way, we update the feature state for the foreground and avoid unnecessary forced adjustments for the background. At the beginning of refinement phase, the input is equal with motion changing feature $x^{l-1} = F^{ref}$. After iterative updating, we get the final refinement frame feature $F_k^{ref} = H^L$, which is decoded to obtain the predicted frame.

3.3 Loss function

Previous works (Wang et al., 2017; Yu et al., 2020; Guen & Thome, 2020) used L_1 or L_2 loss function to train the model. Without the shortcut between lower-layers and higher-layers, the vanishing gradient problem appears not only in the temporal dimension but also between the top layer and bottom layer in the stacked RNN structure. To alleviate this problem, we propose to constrain the hidden state of each layer during the training phase.

In the refine LSTM, the predicted frames are supposed to be clearer as the refine RNN iterative generating. Therefore, we adjust the multi-layer loss by weighted loss function:

$$\mathcal{L}_{\text{pixel}} = \sum_{l=1}^{L} \alpha^{L-l} \left\| x_{t:t+T}^{l} - \hat{x}_{t:t+T}^{l} \right\|_{\tau}, \tag{7}$$

In the motion LSTM, we propose a practical loss to decouple the motion dynamics from spatiotemporal encoding features. Specifically, we provide motion changing area loss for the motion LSTM which ignores the exact pixel value error and focuses on the area changing at adjacent moments. The area A_{t+k} is a binary mask $\{0, 1\}$, which is calculated from the residual of frames $A'_{t+k} = |x_{t+k} - x_{t+k-1}|$ with threshold of θ. The output frames of motion LSTM are supposed to match the ground truth:

$$\mathcal{L}_{area} = \left\| A_{t:t+T} - \hat{A}_{t:t+T} \right\|_{2}. \tag{8}$$

where τ means L_1 and L_2 loss and α^{L-l} is the loss weight for different iterations of the refine LSTM cell. It provides a ground truth guide for both preliminary forecast and final forecast. Compared with previous deep architecture for complex spatiotemporal coherence modeling, our refinement LSTM predicts future frames with explicit intermediate results that are more intuitive.

The overall loss of our CDNet is

$$\mathcal{L} = \lambda_{\text{area}} \mathcal{L}_{\text{area}} + \lambda_{\text{pixel}} \mathcal{L}_{\text{pixel}}, \tag{9}$$

where λ_{pixel} and λ_{area} are adaptive weights.

4 Evaluation

4.1 Datasets

We evaluate our CDNet on two challenging datasets: Human 3.6M dataset (Ionescu et al., 2013) and UCF101 dataset (Soomro et al., 2012).

Human 3.6M dataset is a human motion analysis dataset, which contains 15 kinds of motions acted by 11 actors in a stationary background. The RGB data in human 3.6m dataset is used for the video prediction task. The same with previous work (Wang et al., 2019), all images are cropped from the center and resized to the resolution of 128 × 128. Both in the training phase and prediction phase, 4 future frames are predicted based on the past 4 frames. We use a variety of evaluation metrics to measure the prediction quality, including the Mean Square Error (MSE), Structural Similarity Index Measure (SSIM), Peak Signal to Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), and the Mean Absolute Error (MAE). All the metrics are frame-wise. To facilitate reading, we changed the order of magnitude of the indicators in the following tables. That is, MSE*, LPIPS*, and MAE* are used to represent $\mathrm{MSE}/10$, $\mathrm{LPIPS}*1000$, and $\mathrm{MAE}/100$, respectively.

UCF101 dataset is originally an action recognition dataset collected from YouTube. Recently, considering its rich action categories and complex background, it is gradually used for the video prediction task. There are various pre-processing ways for UCF101 in previous work. In this work, we evaluate our method in two settings. One is that the resolution is reduced to 64 × 88 and 10 future frames are predicted based on the past 10 frames. The other is that the resolution is reduced to 160 × 120 and 10 future frames are predicted based on the past 4 frames. We sub-sample each video by a factor of two, and use the same evaluation metrics with human 3.6m dataset.

Table 1: Results on the Human3.6M dataset. Lower MSE, MAE, LPIPS scores and higher SSIM, PSNR scores mean better results.

Method	MSE*(↓)	SSIM(↑)	PSNR(↑)	LPIPS*(↓)	MAE*(↓)
ConvLSTM (Xingjian et al., 2015)	50.4	0.776	-	-	18.9
FRNN (Oliu et al., 2018)	49.8	0.771	-	-	19.0
MIM (Wang et al., 2019)	42.9	0.79	-	-	17.8
PredRNN (Wang et al., 2017)	48.4	0.781	-	-	18.9
MotionRNN (Wu et al., 2021)	34.2	0.846	-	-	14.8
PhyDNet (Guen & Thome, 2020)	22.1	0.903	24.5	11.2	13.6
CrevNet (Yu et al., 2020)	19.6	0.921	25.5	8.8	9.9
CDNet	**15.9**	**0.936**	**26.5**	**7.4**	**8.3**

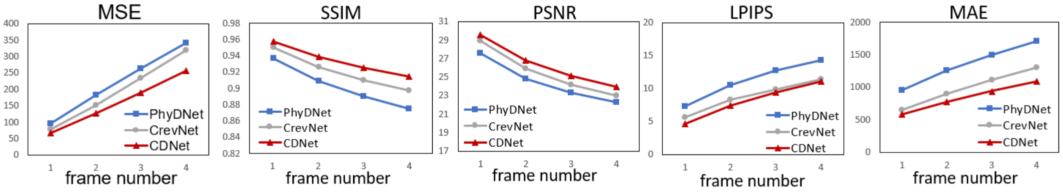


Figure 3: Given 4 frames on the human 3.6m dataset, we specifically show the prediction results of each frame predicted by the model in the future 4 frames. All the referenced models are trained by their open-source code.

4.2 Architecture setup

Our Cascaded Decoupled Network (CDNet) is composed of one layer motion LSTM and 5 layers of the refine LSTM. The dimension of the hidden state is set as 32 for each frame feature representation. The encoder and decoder are using the same 3 convolution layers with batch normalization (Ioffe & Szegedy, 2015) and Relu activate function (Glorot et al., 2011) for 4 times. The global information integration consists of 4 convolution layers with Relu, and the updater U consists of 2 convolution layers with Relu. We adopt the ADAM optimizer (Kingma & Ba, 2014) with the initial learning rate set as $5e-4$. The loss weight α is set as 0.8 and L is equal with the number of the refine LSTM layers. λ_{area} and λ_{pixel} are set as $1e-4$ and 1, respectively. The threshold of area loss is set as 0.05 for normalized pixel value. Our CDNet is implemented in PyTorch (Paszke et al., 2019). More details can be found in Appendix A.

4.3 Results on Human 3.6M dataset

Table 1 summarizes the quantitative results of state-of-the-art methods and our CDNet on the human 3.6m dataset. We can see that our CDNet achieves the best performance on both accuracy metrics (MSE, MAE) and human sensory metrics (SSIM, PSNR, LPIPS), which experimentally proves that our cascaded decoupling structure is effective in modeling motion dynamics and appearance information.

Qualitative Results. Fig. 4 shows some samples of the predicted frames and the residual map between the predicted frames and the ground truth. It can be seen that the PhyDNet (Guen & Thome, 2020) and CrevNet (Guen & Thome, 2020) cannot hold the background information and gradually forgetting appearance details over time. The generated frames are blurry in both the static area and motion area. While our CDNet can predict more reasonable motion dynamics in video frames. In the residual map, our CDNet generates less noise in the background and has low values in the foreground, which means that the CDNet can separate the static region from the motion region, and can syntheses vacant backgrounds and the foreground by referring to global appearance information. Both the quantitative and qualitative results prove the rationality of our architecture. More qualitative results can be found in Appendix B.

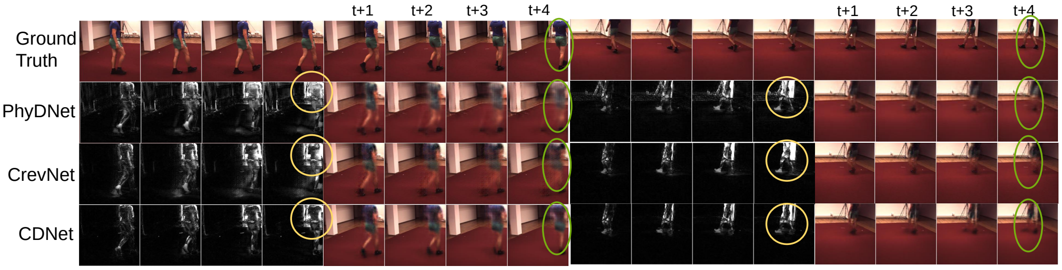


Figure 4: Qualitative results on Human 3.6M dataset.

Table 2: Ablation study on the human3.6m dataset. ”$L1\&L2$” means that we use $L1\&L2$ to constrain the refined forecast of the last iteration. ”without G” means that we remove the global appearance information during the refinement phase. ”without mask” means that the predicted hidden state is directed used in next refinement cell. ”without area loss” means that we train the network without restriction for the motion LSTM.

Method	MSE*(↓)	SSIM(↑)	PSNR(↑)	LPIPS*(↓)	MAE*(↓)
$L1\&L2$	20.1	0.916	25.2	9.3	10.6
without G	17.0	0.928	26.1	8.9	9.7
without mask	16.3	0.928	26.3	9.0	9.7
without area loss	18.1	0.924	25.7	9.3	10.3
CDNet	**16.0**	**0.934**	**26.4**	**8.1**	**8.5**

Ablation study. We evaluate the effect of each module of our CDNet on human 3.6m dataset, and show the Ablation study results in Table 2. It can be seen that the weighted loss and area loss contribute a lot to the improvement of performance, and the global information integration structure and mask structure in the refine LSTM can further improve the performance.

Intermediate representation. To explore the feature representations after decoupling, we decode each predicted feature representation into a visual frame. As shown in Fig. 5, we unpack the generative process at every iteration. It can be seen that the overall trend of numerical results and visualization results is getting better. The image generated by the motion LSTM shows the movement trends, including direction and position, and looks similar to the previous frame in appearance. Later in the refinement process, the human body parts are gradually legible in the predicted position and the missing areas are recovered.

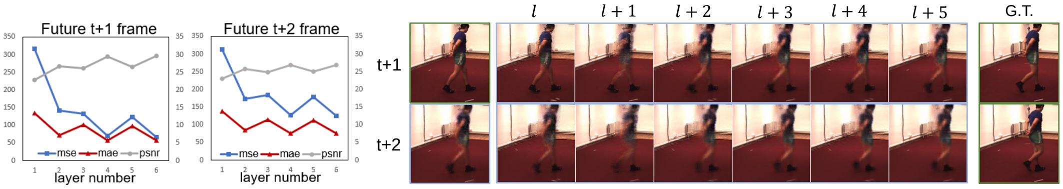


Figure 5: Intermediate representation on human3.6M dataset. (left) MSE, MAE and PSNR results in 1 motion prediction layer and 5 refinement layers at $t+1$ frame prediction and $t+2$ frame prediction. (right) Visualization of the CDNet intermediate prediction. The output in l is the prediction results of the Motion LSTM. Larger visualization can be found in Appendix B.

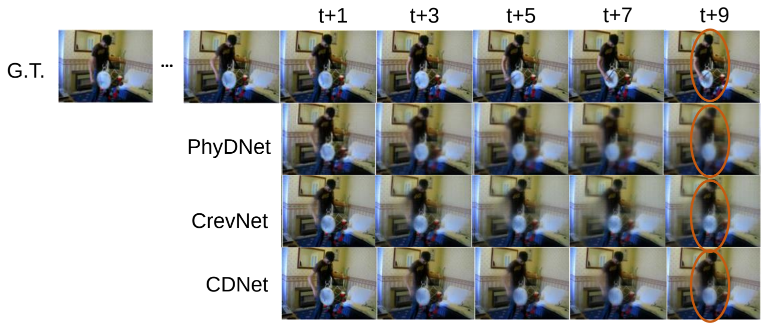


Figure 6: Qualitative results on UCF101 dataset.

4.4 UCF101

Table 3 shows the experimental results on the UCF101 dataset. The results of FRNN is obtained in (Oliu et al., 2018) and we convert the Structural Dissimilarity (DSSIM) in their paper into SSIM by $\mathrm{SSIM}(x, y) = 1 - 2\,\mathrm{SSIM}(x, y)$. It can be seen that our CDNet achieves comparable results in low-resolution setting and better performance than state-of-the-art methods in more challenging high-resolution setting. In the case of low resolution, CrevNet prefers to take the last frame of observation as the future frame, so it has lower LPIPS error and higher MSE error.

Table 3: Results on the UCF101 dataset. We test the CDNet in two settings, including predicting 10 frames based on the past 10 frames at 64 × 88 resolution and predicting 10 frames based on the past 4 frames at 160 × 120 resolution.

Method	MSE*(↓)	SSIM(↑)	PSNR(↑)	LPIPS*(↓)	MAE*(↓)
			10→10		
FRNN (Oliu et al., 2018)	**14.82**	0.74	**23.87**	-	-
PhyNet (Guen & Thome, 2020)	18.85	0.74	21.92	23.62	10.54
CrevNet	23.95	0.74	22.00	**7.29**	9.44
CDNet	16.48	**0.75**	22.06	21.37	**9.31**
			4→10		
PhyNet (Guen & Thome, 2020)	72.56	0.73	20.91	36.27	36.65
CrevNet (Yu et al., 2020)	78.31	0.72	21.10	25.21	34.68
CDNet	**68.38**	**0.76**	**21.75**	**24.21**	**31.12**

Qualitative Results. Fig. 6 shows the qualitative high-resolution video prediction results of PhyDNet, CrevNet, and our CDNet. We can see that our CDNet can predict more details in both foreground and background, and the predicted frames are more similar to the ground truth, which demonstrates the strong generalization ability of our CDNet on real-world large datasets.

5 Conclusion

In this paper, we propose a cascaded network architecture for video prediction by decoupling motion dynamics estimation and appearance refinement into two phases. In the motion dynamics estimation phase, the proposed direct highway structure and the motion changing area loss can effectively model the temporal motion dynamics. While in the appearance refinement phase, the proposed refine LSTM cell and the weighted loss function can iteratively refine the predicted frame by referring to the global foreground and background information. Our method achieves better quantitative and qualitative results than state-of-the-art methods on two challenging datasets, which demonstrates the effectiveness of the proposed cascaded network architecture.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

H. K. Chiu, E. Adeli, and J. C. Niebles. Segmenting the future. *IEEE Robotics and Automation Letters*, PP(99):1–1, 2020.

Chelsea Finn, I. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, 2016.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323. JMLR Workshop and Conference Proceedings, 2011.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11474–11484, 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4554–4563, 2020.

Chenchen Jing, Yuwei Wu, Xiaoxun Zhang, Yunde Jia, and Qi Wu. Overcoming language priors in vqa via decomposed linguistic representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11181–11188, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Y. Kwon and M. Park. Predicting future frames using retrospective cycle gan. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1811–1820, 2019a.

Y. Kwon and M. Park. Predicting future frames using retrospective cycle gan. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1811–1820, 2019b.

Wonkwang Lee, Whie Jung, Han Zhang, Ting Chen, Jing Yu Koh, Thomas Huang, Hyungsuk Yoon, Honglak Lee, and Seunghoon Hong. Revisiting hierarchical approach for persistent long-term video prediction. *arXiv preprint arXiv:2104.06697*, 2021.

Zhihui Lin, Maomao Li, Zhuobin Zheng, Yangyang Cheng, and Chun Yuan. Self-attention convlstm for spatiotemporal prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11531–11538, 2020.

Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 716–731, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro. Sdc-net: Video prediction using spatially-displaced convolution. *Springer, Cham*, 2018.

Xingjian Shi, Zhourong Chen, Hao Wang, D. Yeung, W. Wong, and W. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *ArXiv*, abs/1706.08033, 2017.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 879–888, 2017.

Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, pp. 5123–5132. PMLR, 2018a.

Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International conference on learning representations*, 2018b.

Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9154–9162, 2019.

Haixu Wu, Zhiyu Yao, Jianmin Wang, and Mingsheng Long. Motionrnn: A flexible model for video prediction with spacetime-varying motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15435–15444, 2021.

Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5539–5548, 2020.

SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pp. 802–810, 2015.

Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. 2020.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

A Structure details and Extended quality evaluation

Structure of Information Integration. In Section 4.2, we list the number of CNN layers of information integration and updater U in the refine LSTM. Here we show the detailed structure in Fig. 7. The encoded features of observed video frames are stored in memory through a stacked structure $\{I_1, I_2, ..., I_t\}$ to save all available appearance information for both foreground and background. Since shooting camera and moving objects only have small offsets on adjacent frames, the stored appearance information is redundant with each other. We alternately leverage 1x1 convolution kernels and 3x3 convolution kernels to reduce dimensionality and enhance spatial features. The information integration module integrates the encoded features into a unified global appearance feature G. During the prediction phase, refine LSTM refer the global appearance information at each iteration by the 2-layer CNN updater U.

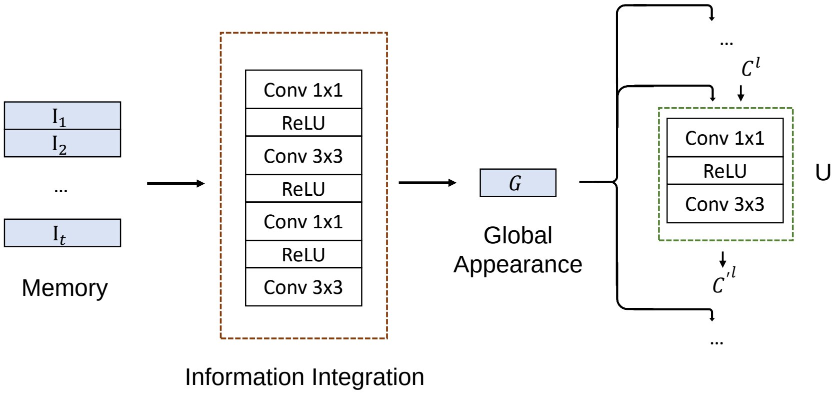


Figure 7: Details of the global information integration and updater U in refine LSTM.

Whole architecture setting. Encoder-Decoder module is directed implemented by the source code of invertible two-way autoencoder in (Yu et al., 2020). The motion LSTM is implemented by the PredRNN (Wang et al., 2017) with the hidden state size, cell state size, and memory state size of 32. In refine LSTM, the hidden state size, cell sate size, memory state size, and mask sate size are also 32. We initialize these states by mapping the encoding state of the first observed frame. During the training phase, we use scheduled sampling to guide the model training, in which we set a probability of 90% to use the ground truth for the next frame predicting, otherwise use the predicted frame of the current time step. This probability decreases exponentially by 0.9 every five epochs.

Additional ablation study. In this section, we test the effect of hyper-parameters on the prediction results. Table 4 shows the prediction results with different iteration number. When the number of iterations is too small, it is difficult for the model to gradually improve the prediction effect. The predictive ability of the model is also related to the dimension of hidden state. Appropriate dimensions can not only reduce computations and memory but also effectively update motion dynamics and appearance state. In addition, we compare the parameters of CrevNet and our CDNet in Table 5. In the case of using 3d convolution, compared with them, we reduce the parameters by 50%. In Fig. 8, we demonstrate the generalization ability of our model. The model is still trained by generating the next 4 frames based on the past 4 frames. During the test, we compare the 8-frame prediction results with PhyDNet (Guen & Thome, 2020), and CrevNet (Yu et al., 2020). In the first 4 frames, our method obviously predicts frames more accurately, and in the next 4 frames, our model still maintains stable prediction. This proves the generalization ability of our model for unlimited length video prediction.

Table 4: Ablation stydy about super-parameters

model		MSE*(↓)	SSIM(↑)	PSNR(↑)	LPIPS(↓)	MAE*(↓)
refine iterations	3	16.6	0.929	26.2	8.9	9.6
	4	18.5	0.911	25.3	10.4	11.8
	5	**15.9**	**0.936**	**26.5**	**7.4**	**8.3**
dimension of hidden state	16	20.5	0.891	24.7	13.6	13.7
	32	**15.9**	**0.936**	**26.5**	**7.4**	**8.3**
	64	16.7	0.930	26.2	8.0	9.6

Table 5: Parameters comparison of CrevNet and CDNet. Parameters are counted by predicting future 4 frames based on the past 4 frames on Human 3.6m.

	Parameters (×10e7)	▽
CrevNet	4.70	-
CDNet	2.31	50.73%

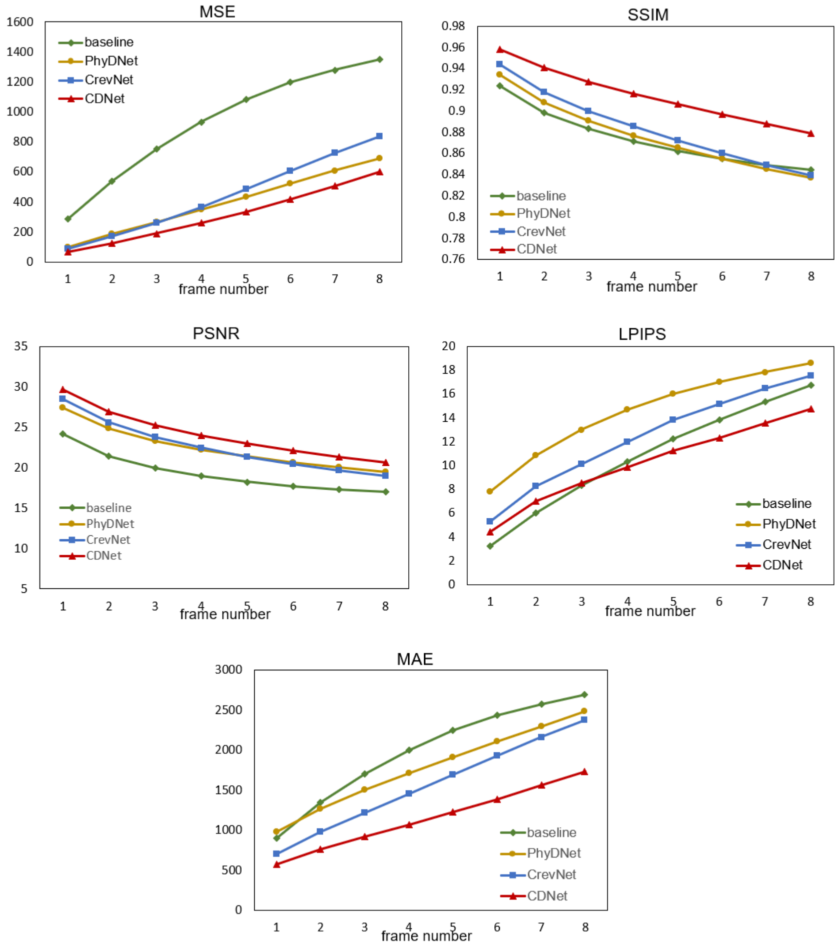


Figure 8: Prediction of 8 future frames based on the model trained by 4→4 setting.

B More qualitative evaluation on Human3.6M dataset and UCF101 dataset

We provide more qualitative results in this section due to the page restriction of the main paper.

Preprocessing of Human3.6M dataset. Human3.6M dataset consists of 15 kinds of actions. The same with previous works (Wang et al., 2019), We just use ”walking” for the video prediction task. The original resolution of the frames is 1000 × 1000 × 3. They are center cropped to 500 × 500 × 3 and resized to 128 × 128 × 3. From the total 7 subjects, we select the S1, S5, S6, S7, S8 for training and S9, S11 for testing. During training and testing phases, we randomly select the starting frame from the frame sequence, as long as the sequence length is adequate.

Qualitative evaluation on human 3.6m dataset. In addition to the Fig. 4 in main paper, we show more qualitative results in Fig. 10 and Fig. 11. We compare our CDNet with state-of-the-art methods, PhyDNet (Guen & Thome, 2020) and CrevNet (Yu et al., 2020) both in RGB frames and error maps. Fig. 10 and Fig. 11 show the predicted 4 future frames. From the predicted RGB frames, we can see that our method predicts human motion dynamics and appearances excellently, significantly outperforming the other two blurry predictions. From the error map, we can also find that our CDNet maintains the human appearance and restores the background area that appears after human movement, which is a difficult task for previous methods.

Preprocessing of UCF101 dataset. UCF101 dataset consists of 101 actions, and each action contains 25 videos. Each video is segmented into a different number of segments. All videos have a resolution of 320 × 240 pixels. We evaluate our method at two resolutions, including 64 × 88 × 3 and 160 × 120 × 3. In the resolution of 64 × 88 × 3, the same with (Oliu et al., 2018), the videos are randomly split into 9957 training segments and 3363 test segments. The frames are down-sampled for every two steps and resized to 64 × 85 × 3. In order to better apply the Encoder-Decoder structure, we change the resolution to 64 × 88 × 3. For the high resolution of 160 × 120 × 3, this requires the model to generate more details. Considering that the results may be affected by random selection of segments, we select 1-19 videos as training sets and 20-25 videos as test sets. The training set contains 10160 segments and the test set contains 3160 segments. The frames are also down-sampled for every two steps and directly resize to 160 × 120 × 3.

Qualitative evaluation on UCF101 dataset. In addition to the Fig. 6 in the main paper, we provide more qualitative results in Figure 12. We can see that although none of the methods predicted the exact movement, our CDNet preserve more appearance information for the unchanging area.

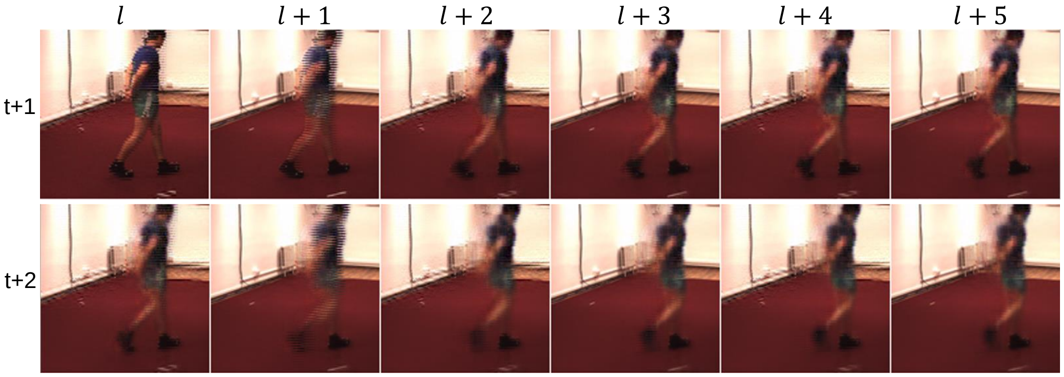


Figure 9: Larger visualization of intermediate representation on human 3.6M dataset, corresponding to Fig. 5 in the main paper.

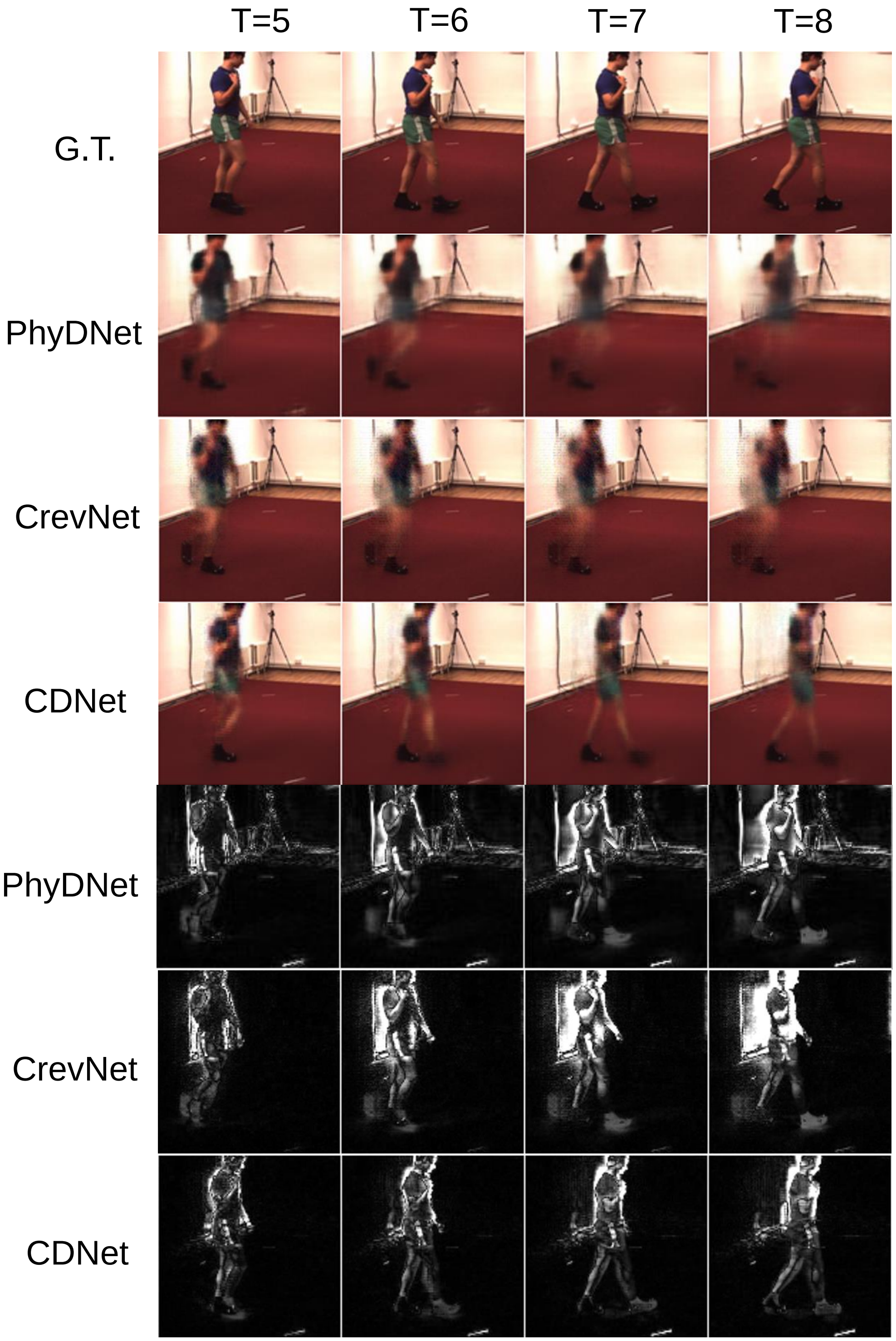


Figure 10: Qualitative comparisons on Human 3.6M. We display predictions of PhyDNet (Guen & Thome, 2020), CrevNet (Yu et al., 2020) and our CDNet starting from the 5th frame to 8th frame. Sample (1).

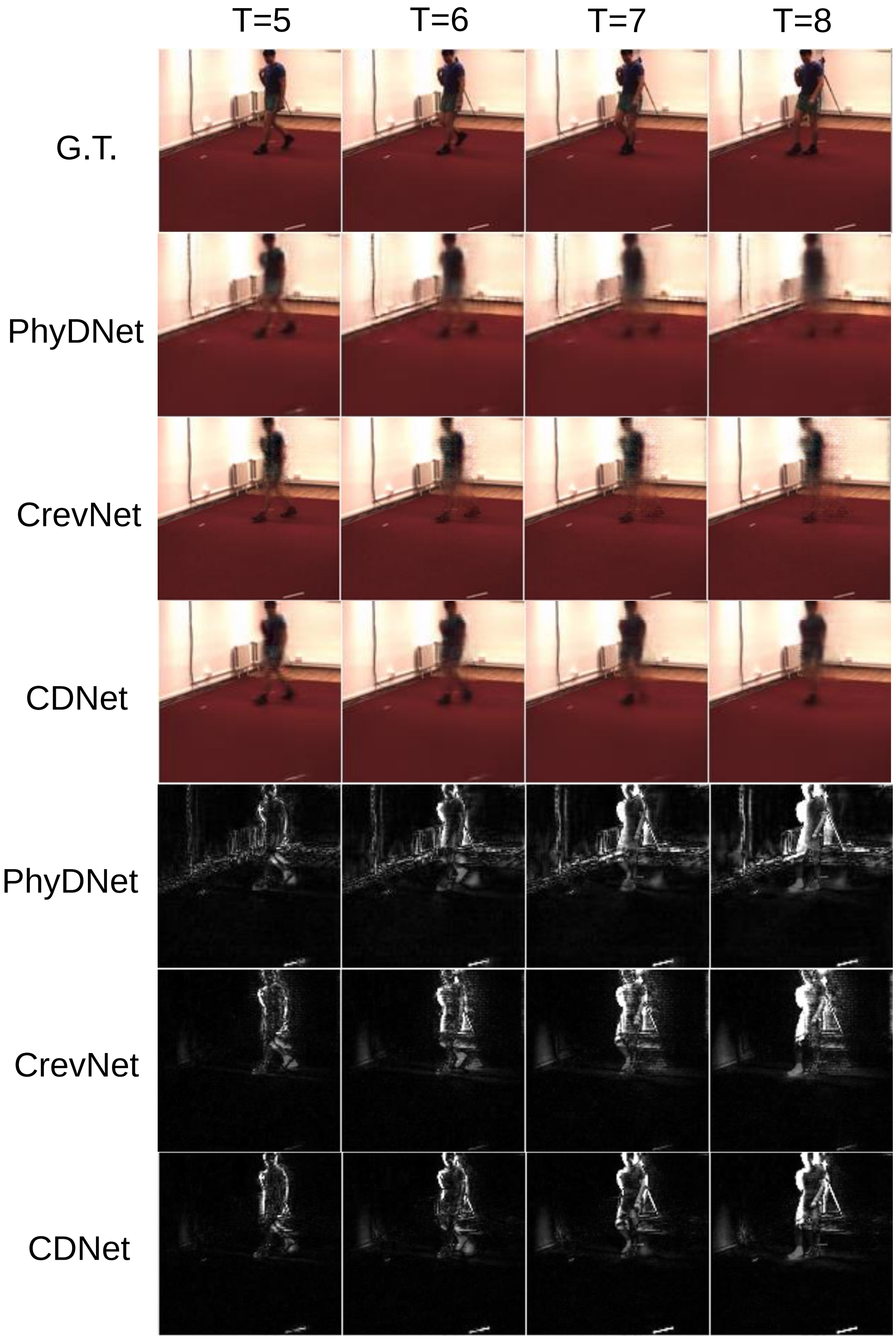


Figure 11: Qualitative comparisons on Human 3.6M. We display predictions of PhyDNet (Guen & Thome, 2020), CrevNet (Yu et al., 2020) and our CDNet starting from the 5th frame to 8th frame. Sample (2).

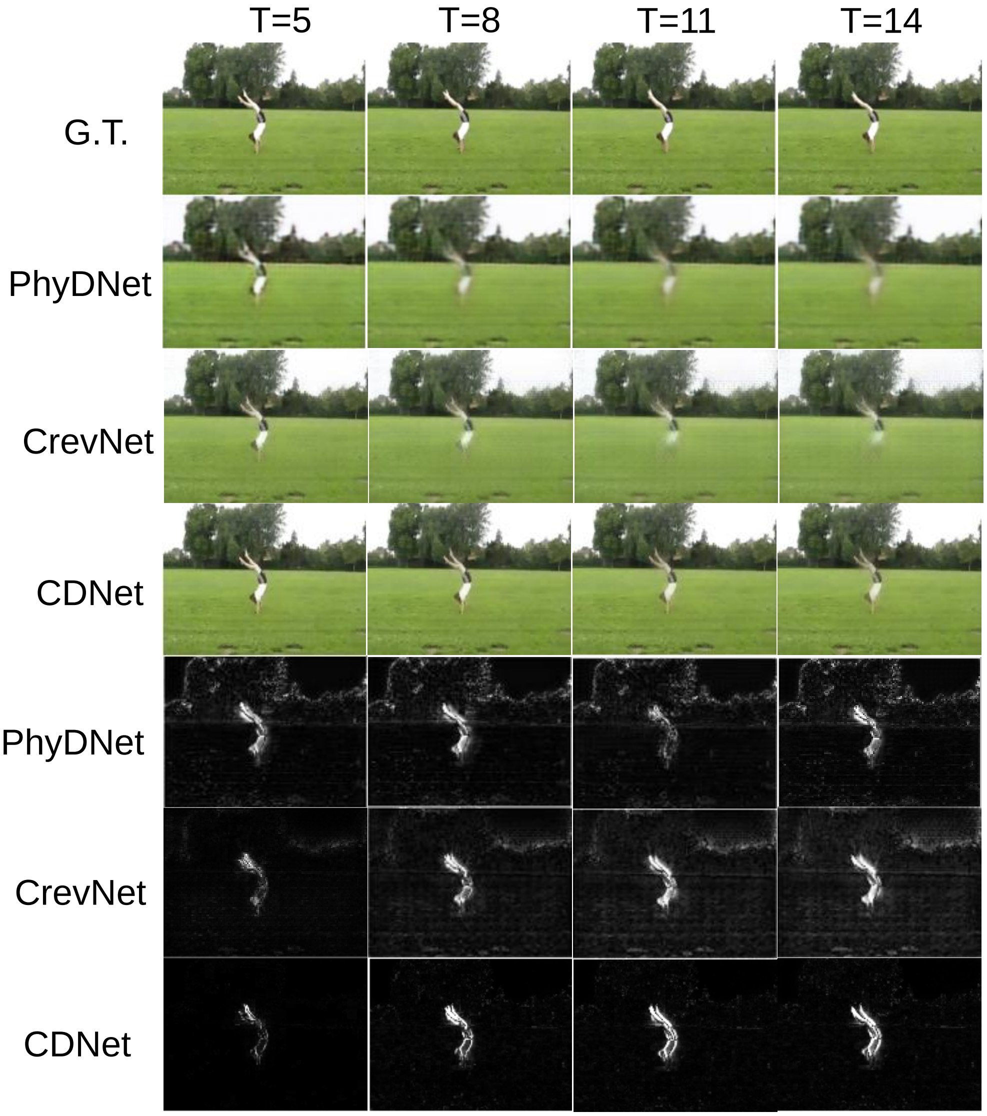


Figure 12: Qualitative comparisons on UCF101. We display predictions of PhyDNet (Guen & Thome, 2020), CrevNet (Yu et al., 2020) and our CDNet starting from the 5th frame to 14th frame, with 3 frames interval. They are trained by past 4 high resolution frames.