DBDNet: Learning Bi-directional Dynamics for Early Action Prediction

Guoliang Pang[1] , **Xionghui Wang**[1] , **Jian-Fang Hu**[1,2*] , **Qing Zhang**[1] and **Wei-Shi Zheng**[1,3]
[1]Sun Yat-sen University, China
[2]Guangdong Province Key Laboratory of Information Security Technology, China
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
{panggliang, wxiongh}@mail2.sysu.edu.cn, hujianf5@mail.sysu.edu.cn, zhangqing.whu.cs@gmail.com, wszheng@ieee.org

Abstract

Predicting future actions from observed partial videos is very challenging as the missing future is uncertain and sometimes has multiple possibilities. To obtain a reliable future estimation, a novel encoder-decoder architecture is proposed for integrating the tasks of synthesizing future motions from observed videos and reconstructing observed motions from synthesized future motions in an unified framework, which can capture the bi-directional dynamics depicted in partial videos along the temporal (past-to-future) direction and reverse chronological (future-back-to-past) direction. We then employ a bi-directional long short-term memory (Bi-LSTM) architecture to exploit the learned bi-directional dynamics for predicting early actions. Our experiments on two benchmark action datasets show that learning bi-directional dynamics benefits the early action prediction and our system clearly outperforms the state-of-the-art methods.

1 Introduction

Predicting human actions from partially observed action sequences is an important research problem with many real-world applications in visual surveillance, human-machine interaction, and medical monitoring, etc. For example, it would be helpful in medical care if the monitoring system equipped in the hospital can forecast the patients' fall. However, action prediction is very challenging, because the observed action information is usually limited, while the future motions to be predicted are highly uncertain.

Many works have been proposed to address this challenge. Some approaches work by mining action cues from the observed video without explicitly modelling the missed future actions [Ryoo, 2011; Hu *et al.*, 2018a]. For example, [Ryoo, 2011] and [Lai *et al.*, 2018] matched a test ongoing video with a set of source videos with ground truth action labels, where the similarity is measured by the distance of the corresponding features. [Hu *et al.*, 2018a] proposed a soft-RNN model to utilize temporal dynamic information for action prediction.

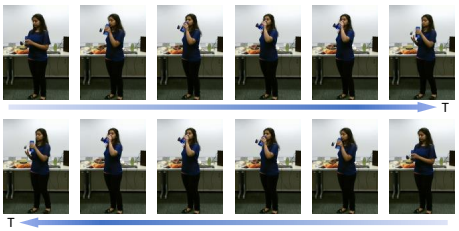


Figure 1. An example action video playing in the normal temporal order (top) and reverse chronological order (bottom). The *bi-directional* sequences consistently indicate that the woman is drinking. The snapshots are from a sample in the NTU RGB+D dataset.

However, these approaches often perform poorly when the observed actions are no more than 10% of the full video, since they all lack of leveraging the contextual information about the future motions.

Other researchers improved the performance of early action prediction by additionally capturing some contextual information about the future motions. These methods basically rely on the discovered temporal ordering dynamics to work. For instance, [Walker *et al.*, 2017] predicted the visual representation of images in the future. [Kong *et al.*, 2017] constructed a mapping to the features of full videos from partial videos. However, a common limitation of these approaches is that the unidirectional temporal dynamics they rely to work is usually noisy and do not contain enough contextual information to achieve reliable action prediction.

To improve the understanding of video action contexts, we propose to analyze the action contexts in a bi-directional manner. As shown in Figure 1, the video playing in two opposite temporal orders both describe that the woman is drinking water, which indicates that: (i) the bi-directional dynamics in action sequence can complement each other; (ii) combining them can obtain more complete action contexts for characterizing human actions. Hence, we argue that the bi-directional temporal contexts could be exploited for obtaining more reliable early action prediction.

Based on the above observations, we develop a novel deep learning framework that takes full advantage of the bi-directional dynamics for early action prediction. Firstly, we introduce a *motion synthesis block* to generate future motions from observed historical motions to relieve the short-

*Corresponding author

age of contextual information. Secondly, we propose a *motion reasoning block* to reconstruct the observed historical actions from the synthesized future motions. It works as a form of regularization to the motion synthesis block, forcing it to produce more reliable future estimation. Finally, we employ an *early action prediction block* to exploit the bi-directional dynamics distilled from the motion synthesis and reasoning blocks for early action prediction. In this way, we can utilize more contextual information than traditional method. All of these blocks form an integrated system for predicting action labels from partially observed videos.

In summary, our contributions are: (i) a novel deep learning framework which learns bi-directional dynamics information in videos for early action prediction; (ii) a complementary encoder-decoder architecture for predicting reliable future motions. Our experiments on two benchmark datasets (UCF 101 and NTU RGB+D action sets) demonstrate that the proposed method can predict actions at early stages and outperform the state-of-the-art by a clear margin on both sets.

2 Related Work

Action Recognition. Action recognition is a long-term research problem and has been studied for decades. Existing methods mainly focus on modelling the temporal dependencies depicted in the observed successive video frames [Simonyan and Zisserman, 2014; Wang *et al.*, 2016; Tran *et al.*, 2015; Carreira and Zisserman, 2017]. For instance, [Li *et al.*, 2018] construct multi-level video representations for by employing an aggregation module at different convolutional layers. [Wang *et al.*, 2016] directly averaged the motion cues depicted in different temporal segments in order to capture some temporal dynamic information. [Tran *et al.*, 2015] employed 3D convolution to explicitly model the temporal relationships. [Hu *et al.*, 2018b] proposed to model the relationship among sequences of varied temporal lengths and modalities by a bi-linear pooling operator. These approaches achieved good performance in several benchmark datasets. However, they are specialized for recognizing actions from full videos and can not be used for predicting partial actions.

Early Action Prediction. Different from action recognition, action prediction is to predict the label of *ongoing actions* based on the partial observation of action executions, which contains less information than the full observation. Recent works have made efforts to recognize actions from partial videos. For instance, [Ryoo, 2011] proposed integral bag-of-words (IBoW) and dynamic bag-of-words (DBoW) to discover some dynamic action evident for prediction. [Lan *et al.*, 2014] divided human actions into multiple levels of granularities and developed a max-margin learning framework to learn a robust hierarchical action representation for action prediction. [Hu *et al.*, 2018a] learned an action predictor from both partial sequences and full sequences by learning a set of soft labels for the sub-sequences of varied progress levels. Recently, [Kong *et al.*, 2017] built a deep model called DeepSCN to learn the connections between full videos and partial videos, such that the algorithm can obtain a feature representation by only observing partial sequences. The DeepSCN was further extended by employing the adversarial learning mechanism to constraining the generation of features for full sequences. However, they did not explore the dynamics among the sub-sequences of different progress levels while our approach is able to dig bi-directional dynamics out along with both temporal order and reverse chronological order and make use of them for early action prediction.

Motion Synthesis. Human motion synthesis aims to generate future motions based on observed incomplete motions. [Fragkiadaki *et al.*, 2015] established an encoder-recurrent-decoder mechanism to learn the dynamics. [Jain *et al.*, 2016] employed structural RNN to mine spatio-temporal interactions. [Ghosh *et al.*, 2017] proposed the Dropout Auto-encoder LSTM (acLSTM) to reduce accumulation of correlated error and thus can capture Long-term dynamics. [Zhou *et al.*, 2018] proposed Auto-Conditioned RNN to fix the problem of error accumulation. More recently, [Tang *et al.*, 2018] introduced an attention module to modify the highway unit in order to capture more motion context.

3 Our Approach

We present a novel deep network (DBDNet) to learn bi-directional dynamics in video for early action prediction. Overall, the workflow of our network is summarized as follows. Firstly, we mine the bi-directional action contexts by a complementary encoder-decoder architecture, which predicts future motions from observed action sequences. Then we reversely reconstruct the observed actions from the predicted future motions. Finally, we feed the mined bi-directional dynamics into a Bi-LSTM[Graves and Schmidhuber, 2005] to recognize ongoing actions. In the following, we elaborate the architecture of our DBDNet.

3.1 Network Architecture for DBDNet

The detailed architecture for DBDNet is presented in Figure 2. As shown, it consists of three blocks: a motion synthesis block encoding the historical motion cues to generate future motions; a motion reasoning block decoding the future motion information to reconstruct the observed historical motions. The motion synthesis block and reasoning block form an encoder-decoder architecture for producing reliable future motions. Finally, an action prediction block is employed for classifying human actions using the bi-directional dynamics mined from the motion synthesis and reasoning blocks. Overall, the combination of these blocks forms a complete system for predicting action labels from partially observed sequences. In the following, we describe each block in detail.

Learning Motion Synthesis: past-to-future

This block is defined such that it can encode long-term dynamics depicted in the observed motion sequences along with the past-to-future direction. It would propagate the contextual information from the historical frames to the future frames. With this block, our system has the ability of predicting long-term future motion descriptors by only observing a part of the motion sequences.

Considering the error accumulation issue in motion synthesis, we formulate the motion synthesis block using an acLSTM model [Zhou *et al.*, 2018]. To train this module, the ground truth g_i and recursive output p_i are fed into the

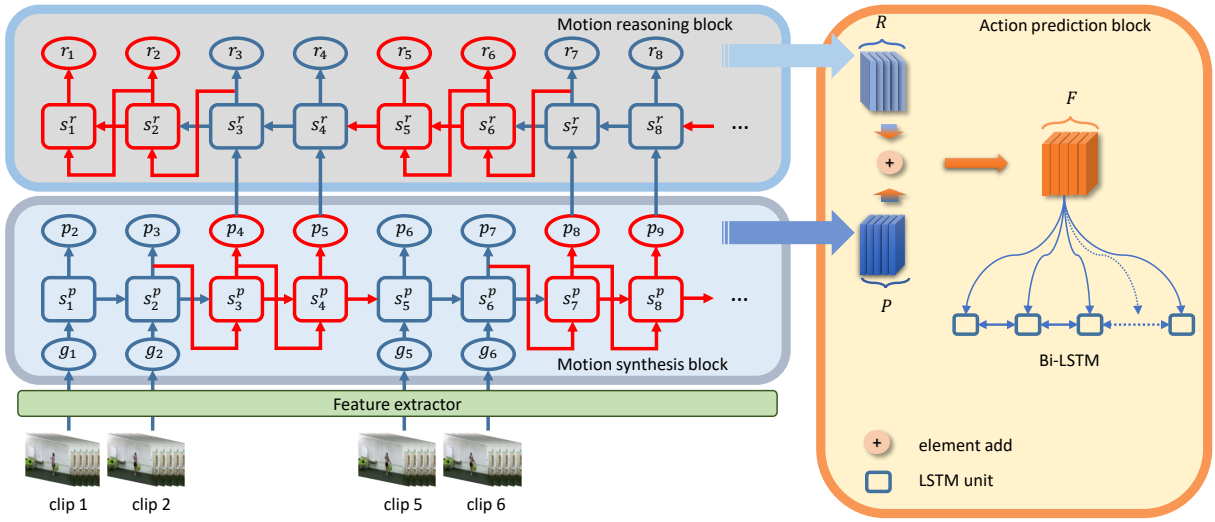


Figure 2. Overall architecture of the proposed DBDNet. For better clarification, the LSTM units with ground truth inputs and recursive inputs are colored with blue and red respectively in this figure. g_t indicates the ground truth input at time step t. s_t^p and s_t^r are the corresponding hidden states. p_t and r_t are outputs of the motion synthesis and reasoning blocks. $\mathbf{P}$ and $\mathbf{R}$ are the sequences of p_t and r_t stacked along the temporal dimension, respectively. $\mathbf{F}$ is the augmented feature generated by element-wisely weighted adding $\mathbf{P}$ and $\mathbf{R}$. When training motion synthesis block, the ground truth g_t (in blue unit) and recursive output p_t (in red unit) are fed into the block in an alternating manner. When training motion reasoning block, the synthesized future frame p_t (in blue unit) and recursive output r_t (in red unit) are used to reconstruct the ground truth g_t in the same way. The mathematical expressions for the motion synthesis block and motion reasoning block can be found in Section 3. The augmented feature F is fed into a Bi-LSTM for predicting the action label of observed videos.

network in an circular fashion, as illustrated in Figure 2. For every u ground-truth inputs, we add v instances of the block's output into its subsequent input streams. We refer u and v as "*ground truth length*" and "*condition length*". Figure 2 is a example with $u = 2$ (in blue) and $v = 2$ (in red). Compared with conventional RNN/LSTM techniques, acLSTM has the following advantages. First, it is conditioned on its own output during training and thus can reduce the issue of error accumulation; Second, it allows the network to encode long-term dynamics. We define the loss of this module as:

$$L_1 = \sum_{i=2}^{T} \|\mathbf{p}_i - \mathbf{g}_i\|_{l_2}^2, \tag{1}$$

where $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_T]$ is the T-length ground truth motion sequence used for model training, which is fed into the synthesis block. $\mathbf{P} = [\mathbf{p}_2, \mathbf{p}_3, ..., \mathbf{p}_{T+1}]$ is the forward dynamic sequences output by the motion synthesis block. $\mathbf{p}_{i+1}$ is given in a recursive form as

$$\begin{aligned} \mathbf{p}_{i+1} &= \mathbf{W}_p E(\mathbf{x}_i^p, \mathbf{s}_i^p), \\ \mathbf{x}_i^p &= \begin{cases} \mathbf{g}_i, & \text{if } mod(i, u+v) \in (0, u] \\ \mathbf{p}_i, & \text{if } mod(i, u+v) \in (u, u+v) \cup \{0\}, \end{cases} \end{aligned} \tag{2}$$

where E denotes the encoder architecture, which is instantiated to a LSTM unit here. $\mathbf{W}_p$ is the corresponding output condition matrix. mod is the modulo operation.

Learning Motion Reasoning: future-back-to-past

We define this block for capturing backward propagated dynamics that allow to reconstruct the historical frames from the the synthesized future frames. The outputs of synthesis block $\mathbf{P}$ are then fed into this block in a reverse chronological order, for reconstructing the observed historical motions. Thus, the contextual information is explicitly propagated from future to history in this module. It allows the network to decode the synthesized future motions to produce an estimation about the observed historical motions. Here, we also construct our motion reasoning block as acLSTM model, which has a dual structure with the motion synthesis block, as illustrated in Figure 2. Similar to the motion synthesis module, we minimize the gap between the reconstructed historical motions and the corresponding ground truth motions. The loss function for this module is

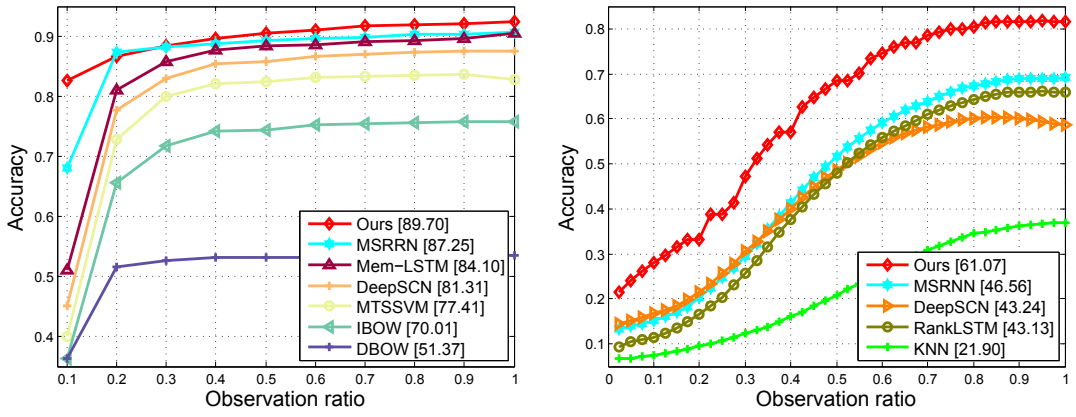
$$L_2 = \sum_{j=1}^{T} \|\mathbf{r}_j - \mathbf{g}_j\|_{l_2}^2, \tag{3}$$

where $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_T]$ stacks the backward dynamics gained by the motion reasoning block. Here, $\mathbf{r}_j$ is given by

$$\begin{aligned} \mathbf{r}_j &= \mathbf{W}_r D(\mathbf{x}_j^r, \mathbf{s}_j^r), \\ \mathbf{x}_j^r &= \begin{cases} \mathbf{r}_{j+1}, & \text{if } mod(j, u+v) \in (0, u] \\ \mathbf{p}_{j+1}, & \text{if } mod(j, u+v) \in (u, u+v) \cup \{0\}, \end{cases} \end{aligned} \tag{4}$$

where D indicates the decoder architecture, which is implemented as a LSTM unit here. $\mathbf{W}_r$ is the output condition matrix for the motion reasoning block. Here, $\mathbf{r}_j$ is recursively computed in the reverse chronological direction.

It is worth noting that this block can also serve as a constraint to the motion synthesis block. Intuitively, a bad prediction of future motions would lead to large loss in the reasoning block, which would guide the algorithm to adjust the parameters of synthesis block. The combination of motion synthesis block and reasoning block forms an encoder-decoder



(a) UCF101 dataset (b) NTU RGB+D dataset

Figure 3. Prediction results on the (a) UCF101 dataset and (b) NTU RGB+D action dataset.

mechanism for producing reliable future motions. It can capture rich bi-directional dynamics in observed videos, which would benefit the prediction of early actions.

Bi-directional Early Action Prediction

Here, we introduce an action prediction block to exploit the bi-directional dynamics for early action prediction. Specifically, we first fuse the features output by the motion synthesis and reasoning blocks to form an augmented feature representation for the observed partial videos, which is defined as $\mathbf{F} = \alpha\mathbf{P} + (1-\alpha)\mathbf{R}$. Then, we feed the augmented feature $\mathbf{F}$ into a Bi-LSTM architecture to mine more discriminative bi-directional dynamics for early action prediction. Bi-LSTM can use both past and future context in each position, in favor of better utilization of our bi-directional outputs. In this block, a standard cross-entropy loss (denoted by L_3) is employed to guide the learning of our prediction block.

Loss Function

We use all the videos (partial and full) to train our DBDNet for early action prediction. Our objective is to minimize the following loss function:

$$L = \sum_{i=1}^{T}(L_1^i + w_1 L_2^i + w_2 L_3^i), \tag{5}$$

where w_1 and w_2 are parameters to control the contribution of different losses. L_*^i indicates the training loss of the corresponding block on partial videos with observation ratio i/T.

3.2 Model Optimization

It is not easy to directly optimize the DBDNet. Here, we describe a two-step method to determine the model parameters. We empirically found that optimizing DBDNet in this way can obtain a better performance for early action prediction, which will be further discussed in the ablation study section.

Step-1. In this step, we pre-train the parameters of the motion synthesis and reasoning blocks. Here, we freeze the action prediction block and directly minimize loss $L_1 + w_1 L_2$ over the parameters of the motion synthesis and reasoning blocks. Following the implementation in [Zhou *et al.*, 2018], we feed the full videos into the motion synthesis and reasoning blocks to train the acLSTMs.

Step-2. In the step-2, we tune all the parameters involved in the DBDNet model by making use of all the partial and full action sequences. Here, we set the parameters "*ground truth length*" and "*condition length*" as the lengths of observed videos and future videos to be predicted, respectively. This means that the acLSTM architectures of the motion synthesis and reasoning blocks degenerate into a conventional LSTM in this step, whose parameters remain the same as the corresponding acLSTM. We then jointly train the motion synthesis and reasoning blocks together with the action prediction block by minimizing the loss function defined in Eq. (5).

4 Experiments

We conducted experiments on two benchmark sets for early action prediction and compared our method with state-of-the-arts. In the following, we first describe the implementation details and parameter setting, and then report the results.

4.1 Implementation Details

We instantiated motion synthesis block and motion reasoning block as an one-layer acLSTM with a fully connected layer. We defined the action prediction block as an one-layer Bi-LSTM. The weight α for fusing the outputs of motion synthesis and reasoning block was set as 0.6 in all of our experiments. Its effect would be studied in the ablation study. The parameters "condition length" and "ground-truth length" in acLSTMs were set as 1. We set the hidden sizes of acLSTM and Bi-LSTM as 2048 and 768, respectively. We placed a dropout layer on top of Bi-LSTM, where the probability was set as 0.5. We optimized our DBDNet using Adam algorithm with a batch size of 32 in all of our experiments.

For the experiments on UCF101 set, we followed the settings in [Kong *et al.*, 2017] and [Hu *et al.*, 2018a] and partitioned each video into 10 shorter segments. We used the 3D

Observation ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	AUC
DBoW	36.29	51.57	52.71	53.13	53.16	53.24	53.24	53.34	53.45	53.53	51.37
IBoW	36.29	65.69	71.69	74.25	74.39	75.23	75.36	75.57	75.79	75.79	70.01
MTSSVM	40.05	72.83	80.02	82.18	82.39	83.21	83.37	83.51	83.69	82.82	77.41
DeepSCN	45.02	77.64	82.95	85.36	85.75	86.70	87.10	87.42	87.50	87.63	81.31
Men-LSTM	51.02	80.97	85.73	87.76	88.37	88.58	89.09	89.38	89.67	90.49	84.10
MSRNN	68.00	**87.39**	88.16	88.79	89.24	89.67	89.85	90.28	90.43	90.70	87.25
Ours	**82.67**	86.61	**88.35**	**89.71**	**90.58**	**91.12**	**91.69**	**91.85**	**92.02**	**92.40**	**89.70**

Table 1. Comparison results (%) on the UCF101 dataset.

Observation ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	AUC
KNN	7.45	9.56	12.25	16.04	20.89	25.97	30.85	34.49	36.15	37.02	21.90
RankLSTM	11.54	16.48	25.66	37.74	47.96	55.94	60.99	64.41	66.05	65.95	43.13
DeepSCN	16.80	21.46	30.51	39.93	48.73	54.61	58.18	60.18	60.01	58.62	43.24
MSRNN	15.17	20.33	29.53	41.37	51.64	59.15	63.91	67.38	68.89	69.24	46.56
Ours	**27.98**	**33.30**	**47.27**	**56.94**	**68.54**	**74.50**	**78.53**	**80.51**	**81.63**	**81.54**	**61.07**

Table 2. Comparison results (%) on the NTU RGB+D dataset.

ResNext-101 network [Hara *et al.*, 2018] pre-trained on Kinetics dataset [Kay *et al.*, 2017] without finetuning[1] to extract visual features. The learning rate was set as 1×10^{-5} for both the motion synthesis and reasoning blocks, and 5×10^{-6} for the action prediction block. The parameters w_1 and w_2 were set as 1 and 0.01, respectively.

For the experiments on NTU RGB+D action set, we kept the same evaluation settings as [Hu *et al.*, 2018a] and uniformly divided each full sequence into 40 shorter segments. We then extracted visual features from the RGB and depth channels by training two 16-channel-InceptionResNetV2 networks. We also extracted skeleton features using a 3-layer LSTM network. All the three features were then concatenated together and fed into our DBDNet. The learning rate was set as 1×10^{-4} for both motion synthesis and reasoning blocks, and 1×10^{-3} for the action prediction block. The parameters w_1 and w_2 were set as 1 and 0.1 respectively.

4.2 Results for Early Action Prediction

In the following, we report and discuss our experimental results on the UCF101 and NTU RGB+D action sets.

UCF101 Dataset

The UCF101 dataset consists of 13,320 videos from 101 action categories. These videos are divided into 25 groups. Following the evaluation criterion in [Kong *et al.*, 2018], we used the videos from the first 15 groups for training, the next 3 groups for validation, and the last 7 groups for testing.

We compare our method with approaches including Integral BoW (IBOw), Dynamic BoW (DBoW)[Ryoo, 2011], MTSSVM[Kong *et al.*, 2014], DeepSCN[Kong *et al.*, 2017], Mem-LSTM[Kong *et al.*, 2018], and MSRNN[Hu *et al.*, 2018a]. The detailed comparison results are presented in Figure 3(a) and Table 1. As shown, our method obtains the best prediction results on this set and outperforms the state-of-the-art MSRNN by a margin of 2.45% in the term of area under curve (AUC). Particularly for the actions at very early stages (e.g., less than 20%), our approach has distinct advantage, which demonstrates the effectiveness of learning the bi-directional dynamics in video for early action prediction. By exactly examining the detailed prediction results in Table 1, we can observe that our method can consistently outperform other approaches on most of the progress levels. It is worthy noting that our method also outperforms the Mem-LSTM [Kong *et al.*, 2018] by more than 5%, which only utilized the unidirectional temporal dynamics to represent actions. We are pleased to see that our method can obtain an accuracy of 90.58% for the prediction of partial actions with an observation ratio of 50%.

NTU RGB+D Action Dataset

The NTU RGB+D action dataset contains 56,880 RGB+D videos from 60 actions. All of the actions in this set are performed by 40 subjects for several times and captured from different views. This set is very challenging for early action prediction as actions in this set often contain very similar action contents at early stages. For experiments, we employed the cross-subject evaluation protocol described in [Hu *et al.*, 2018a], where a subset with 40,320 action samples are used for training and the remaining 16,560 samples for test.

We compare our results with KNN[Hu *et al.*, 2018a], RankLSTM[Ma *et al.*, 2016], DeepSCN[Kong *et al.*, 2017], and MSRNN[Hu *et al.*, 2018a]. The comparison results are presented in Figure 3(b) and Table 2. As can be seen, our DBDNet model can achieve an impressive prediction performance on this set, with an AUC of 61.07%, which outperforms the state-of-the-art by a large margin (about 14%).

5 Ablation Study

In the following, we evaluate the effectiveness of the pre-training of acLSTM (i.e., step-1 in the model optimization), the influence of parameters α, w_1, and w_2. We also study the effect of the reasoning block in our DBDNet. All the experiments are conducted on the NTU RGB+D action dataset.

[1] https://github.com/kenshohara/3D-ResNets-PyTorch

Observation ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	AUC
no-Reasoning	26.80	31.88	44.66	54.15	66.06	72.84	77.36	79.56	81.43	**81.81**	59.56
Ours	**27.98**	**33.30**	**47.27**	**56.94**	**68.54**	**74.50**	**78.53**	**80.51**	**81.63**	81.54	**61.07**

Table 3. System performances (%) with vs. without motion reasoning block in the DBDNet framework.

Observation ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	AUC
No-Pretrain	26.12	31.14	43.47	52.96	65.23	71.65	76.20	78.38	80.73	81.20	58.64
No-acLSTM	26.27	31.93	44.65	54.55	66.39	72.93	77.25	79.65	81.54	**81.60**	59.62
Ours	**27.98**	**33.30**	**47.27**	**56.94**	**68.54**	**74.50**	**78.53**	**80.51**	**81.63**	81.54	**61.07**

Table 4. System performances (%) with vs. without pre-training motion synthesis and acLSTM on the NTU RGB+D dataset.

Evaluation on the motion reasoning block. To evaluate the effectiveness of the reasoning block, we implement a baseline by removing it from our DBDNet framework and test the performances. We denote this baseline by no-Reasoning. The results for the prediction of partial videos under varied observation ratio are presented in Table 3. As can be seen, our method outperforms no-Reasoning over most of the observation ratios, which demonstrates the effectiveness of our reasoning block for DBDNet framework. This also indicates that explicitly learning the backward propagated dynamic from the synthesized future motions to the historical motions is beneficial for early action prediction.

Evaluation on the acLSTM. We have proposed a two-step approach for optimizing our DBDNet in Sec. 3.2. In step-1, we used acLSTMs to pre-train the parameters of motion synthesis and reasoning blocks. Here, we test its influence and implement a baseline directly training DBDNet using step-2. We denote the baseline as No-Pretrain. To validate the effectiveness of acLSTM, we also replace it with a vanilla LSTM and denote this baseline as No-acLSTM. The comparison results are reported in Table 4. We can observe that the AUC will drop to 58.64% from 61.07% when discarding step-1, which manifests that pre-training acLSTMs is beneficial for our model. The reason is that pre-training acLSTMs can provide a proper initialization for the training of motion synthesis and reasoning blocks in step-2. We also see that replacing acLSTM with vanilla LSTM gets worse results (61.07% vs. 59.62%). This is because that acLSTM has the advantage of reducing error accumulation and can mine more reliable dynamic contexts for achieving better performance.

The influence of α. In our DBDNet, we have employed a weight α to fuse the forward and backward propagated dynamics gained by the motion synthesis and reasoning blocks, respectively. Here, we study the sensitivity of our method to it. The results are tabulated in Table 5. When α is set to 1, only the synthesis block is used for prediction (60.28%). And when α is set to 0, only the reasoning block is employed (60.84%). The results show that the forward and backward propagated dynamics can complement with each other to obtain a robust feature representation for the ongoing actions. And a proper combination (e.g., $\alpha = 0.6$) of them can produce better results (61.07%), which confirms the necessity of exploiting bidirectional dynamics for the early action prediction. Note that 1% improvement means that about 160×40 partial videos are correctly predicted in NTU RGB+D set.

α	0	0.2	0.4	0.6	0.8	1.0
AUC	60.84	60.76	61.04	**61.07**	60.83	60.28

Table 5. System performances (%) with various α on the NTU RGB+D dataset.

$w_2 = 0.1$	w_1	0.001	0.01	0.1	1	10	100
	Acc	60.41	60.39	60.30	**61.07**	60.52	60.58
$w_1 = 1$	w_2	0.001	0.01	0.1	1	10	100
	Acc	60.49	60.36	**61.07**	60.39	60.55	60.45

Table 6. System performances (%) with various w_1 and w_2 on the NTU RGB+D dataset.

The influence of w_1 and w_2. We have employed parameters w_1 and w_2 to control the contribution of the losses for the motion synthesis, motion reasoning, and action prediction blocks. Here, we study their influence. To evaluate the influence of w_1, we fix w_2 as 0.1 and set w_1 as 0.001, 0.01, 0.1, 1, 10, and 100, respectively. For investigating the effect of w_2, we change it from 0.001 to 100 and keep w_1 fixed. The detailed results are presented in Table 6. As shown, our model is quite robust to w_1 and w_2. A proper combination of the losses gives a better result, generally too small or too large w_1 and w_2 would result in an inferior performance.

6 Conclusion

In this work, we proposed a deep bi-directional dynamics network (DBDNet) for early action prediction. In the proposed framework, we employed two acLSTMs, which form an encoder-decoder mechanism for producing more reliable future motions, to capture bi-directional dynamics in videos. We also placed a Bi-LSTM on top of the acLSTMs to exploit bi-directional dynamics for predicting ongoing actions. Our experimental results on the UCF101 and NTU RGB+D sets have demonstrated the effectiveness of the proposed method.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2018YFB1004903), NSFC (61702567, 61802453, 61628212), SF-China (61772570), Guangdong Natural Science Funds for Distinguished

Young Scholar (2018B030306025), and FY19-Research-Sponsorship-185. Jian-Fang Hu is supported by Opening Project of GuangDong Province Key Laboratory of Information Security Technology(2017B030314131) and CCF-Tencent open research fund. The corresponding author is Jian-Fang Hu.

References

[Carreira and Zisserman, 2017] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, 2017.

[Fragkiadaki *et al.*, 2015] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[Ghosh *et al.*, 2017] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *International Conference on 3D Vision*, 2017.

[Graves and Schmidhuber, 2005] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[Hara *et al.*, 2018] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[Hu *et al.*, 2018a] Jian-Fang Hu, Wei-Shi Zheng, Lianyang Ma, Gang Wang, Jianhuang Lai, and Jianguo Zhang. Early action prediction by soft regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[Hu *et al.*, 2018b] Jian-Fang Hu, Wei-Shi Zheng, Jiahui Pan, Jianhuang Lai, and Jianguo Zhang. Deep bilinear learning for rgb-d action recognition. In *European Conference on Computer Vision*, pages 346–362, 2018.

[Jain *et al.*, 2016] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[Kay *et al.*, 2017] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, 2017.

[Kong *et al.*, 2014] Yu Kong, Dmitry Kit, and Yun Fu. A discriminative model with multiple temporal scales for action prediction. In *European Conference on Computer Vision*, pages 596–611, 2014.

[Kong *et al.*, 2017] Yu Kong, Zhiqiang Tao, and Yun Fu. Deep sequential context networks for action prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1481, 2017.

[Kong *et al.*, 2018] Yu Kong, Shangqian Gao, Bin Sun, and Yun Fu. Action prediction from videos via memorizing hard-to-predict samples. In *AAAI Conference on Artificial Intelligence*, 2018.

[Lai *et al.*, 2018] Shaofan Lai, Wei-Shi Zheng, Jian-Fang Hu, and Jianguo Zhang. Global-local temporal saliency action prediction. *IEEE Transactions on Image Processing*, 27(5):2272–2285, 2018.

[Lan *et al.*, 2014] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. 2014.

[Li *et al.*, 2018] Yang Li, Kan Li, and Xinxin Wang. Deeply-supervised cnn model for action recognition with trainable feature aggregation. In *International Joint Conference on Artificial Intelligence*, 2018.

[Ma *et al.*, 2016] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1942–1950, 2016.

[Ryoo, 2011] Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *IEEE International Conference on Computer Vision*, pages 1036–1043, 2011.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[Tang *et al.*, 2018] Yongyi Tang, Lin Ma, Wei Liu, and Wei-Shi Zheng. Long-term human motion prediction by modeling motion context and enhancing motion dynamics. In *International Joint Conference on Artificial Intelligence*, 2018.

[Tran *et al.*, 2015] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[Walker *et al.*, 2017] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. *CoRR*, abs/1705.00053, 2017.

[Wang *et al.*, 2016] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[Zhou *et al.*, 2018] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *International Conference on Learning Representations*, 2018.