Memory-Augmented Temporal Dynamic Learning for Action Recognition

Yuan Yuan, Dong Wang, Qi Wang
School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL),
Northwestern Polytechnical University, Xi'an 710072, China,
{y.yuan1.ieee, nwpuwangdong, crabwq}@gmail.com

Abstract

Human actions captured in video sequences contain two crucial factors for action recognition, i.e., visual appearance and motion dynamics. To model these two aspects, Convolutional and Recurrent Neural Networks (CNNs and RNNs) are adopted in most existing successful methods for recognizing actions. However, CNN based methods are limited in modeling long-term motion dynamics. RNNs are able to learn temporal motion dynamics but lack effective ways to tackle unsteady dynamics in long-duration motion. In this work, we propose a memory-augmented temporal dynamic learning network, which learns to write the most evident information into an external memory module and ignore irrelevant ones. In particular, we present a differential memory controller to make a discrete decision on whether the external memory module should be updated with current feature. The discrete memory controller takes in the memory history, context embedding and current feature as inputs and controls information flow into the external memory module. Additionally, we train this discrete memory controller using straight-through estimator. We evaluate this end-to-end system on benchmark datasets (UCF101 and HMDB51) of human action recognition. The experimental results show consistent improvements on both datasets over prior works and our baselines.

Introduction

In recent years, video based human action recognition has received increasing attention from the research community (Wang et al. 2011; Simonyan and Zisserman 2014; Tran et al. 2015; Carreira and Zisserman 2017), owing to its great potential value in many real-world applications like surveillance (Lin et al. 2008), abnormal activity detection (Boiman and Irani 2007) and so forth. Unlike recognition in static images, a distinctive aspect of the action recognition in videos is the motion dynamics, which is a crucial factor in addition to visual appearance. The performance of action recognition system depends, to a large extent, on whether the dynamics can be effectively represented and utilized. A key goal of this work is to enhance the model's capacity for learning and capturing the motion dynamics in videos.

Encouraged by the success of Convolutional Neural Network (CNNs) on computer vision tasks such as image classification, many researchers have adopted similar methods for video understanding and human action recognition (Simonyan and Zisserman 2014; Feichtenhofer, Pinz, and Zisserman 2016; Zha et al. 2015; Sun et al. 2015) and achieved remarkable performance on public benchmark datasets. Deep CNNs have been shown to have an extraordinary capacity for learning discriminative representation from raw visual data. The most successful action recognition algorithm, two-stream convents, utilizes two independent CNNs to extract pre-frame features from RGB and optical flow images, followed by simple or strategic pooling across the temporal domain. However, traditional two-stream convnets learn the action representation frame by frame and lack the capacity to model long-range dynamics across the temporal domain. Recurrent Neural Networks (RNNs) is a straightforward choice to exploit the sequential structure in videos. Long Short-Term Memory (LSTM) and its variants have been explored to incorporate frame-level features in several works (Xingjian et al. 2015; Li et al. 2018; Ma et al. 2017). These models tend to work well for actions with short duration and little movement. However, these approaches would incur low accuracy when applied to long-range and complex motions, as verified by our experiments. These challenges motivate us to design an effective and efficient module to (1) capture salient motion dynamics in long-range temporal structure and (2) compose complex motion information across time for distinguishing actions.

In this paper, we propose a novel network structure, named memory-augmented temporal dynamic learning network, where the frame-level feature representation is stored and recalled from an external memory module, to maintain the informative motion dynamics for classifying actions. It can more effectively learn the long-term motion dynamic without increasing the model complexity. Specifically, a video is converted to a sequence of frame-level features by CNN and communicates with external memory according to a memory controller. There are three elements fed into memory controller at every frame, i.e., memory history, context embedding and current frame-level feature. The memory history is constructed by all features in the memory module, and the context embedding is obtained using a Long Short-Term Memory network that skilled in modeling short-range motion dynamics. We introduce a threshold-based selection mechanism to make memory controller output a dis-

crete decision on whether write the current frame-level feature into memory module. After processed every frame features, the final action representation for recognizing actions is obtained by applying average pooling to features in the memory module. At the same time, the straight-through estimator is utilized to make the discrete memory controller trainable in end-to-end manner. We evaluate our model on two benchmark datasets, and outperform our baseline and state-of-the-art performance.

The main contribution of this paper is summarized as follows:

- We design a memory-augmented temporal dynamic learning network for action recognition. An external memory module is attached to CNNs to store salient and informative motion dynamics in videos, and greatly enhances capacity to encode long-term and complex motion dynamics in long-range temporal structure.
- We propose a discrete memory controller, that takes in the memory history, context embedding and current feature as inputs, to control the writing process of the external memory module. This allows efficiently leveraging information from different temporal scale and improves the representation power, as demonstrated by our experiments.
- We extensively evaluate our algorithm on large-scale datasets UCF101 and HMDB51. Our method performs favorably against state-of-the-art action recognition methods and our baseline.

Related Work

Motivated by the impressive performance of CNNs on image classification (Krizhevsky, Sutskever, and Hinton 2012), semantic segmentation (Long, Shelhamer, and Darrell 2015) and other computer vision work (Wang et al. 2018), several recent works have utilized CNN-based architectures for video based human action recognition. Karpathy et al. (Karpathy et al. 2014) directly apply CNNs to extract frame-level features and exploit multiple simple temporal pooling methods, including early fusion, late fusion, and slow fusion. But these approaches only yield a modest improvement over single frame baseline, indicating that motion dynamic information is hard to model by directly pooling spatial features from CNNs.

In view of this, Tran et al. (Tran et al. 2015) propose Convolutional 3D (C3D) and construct a deep C3D neural network with 3D convolution filters and 3D pooling layers, which operate on short video clips over space and time simultaneously. Noticing that 3D kernels only cover a short range of the sequence when filtering video clips, Simonyan et al. (Simonyan and Zisserman 2014) incorporate motion information by training a temporal stream of CNN on optical frames in addition to spatial stream with RGB frame input. By simple fusing probability scores from these two-stream CNNs, the accuracy of action recognition is significantly boosted. Moreover, several attempts have been made to learn subtle spatio-temporal relationships between appearance and motion in order to improve recognition accuracy. Feichtenhofer et al. (Feichtenhofer, Pinz, and Zisserman 2016) study a number of ways of integrating two-stream CNNs spatially and temporally. They propose s spatiotemporal fusion method by generalizing residual networks. Wang et al. (Feichtenhofer, Pinz, and Wildes 2017) introduce the spatiotemporal compact bilinear operator to efficiently fuse spatial and temporal features hierarchically. However, the only several consecutive optical flow frames are fed into temporal stream CNN, so that it cannot capture longer-term motion patterns associated with certain human actions.

To enable the model to learn long-term motion dynamic, Ng et al. (Yue-Hei Ng et al. 2015) take advantage of LSTM to fuse features across a longer temporal range. However, the vanilla LSTM model is not satisfactory for learning the spatial correlations and motion dynamics between the frames. (Xingjian et al. 2015) extends LSTM to ConvLSTM, which replace linear multiplicative operation with spatial 2D convolution, so that it can learn the spatial patterns along the temporal domain. Furthermore, Shikhar et al. (Sharma, Kiros, and Salakhutdinov 2015) implant soft attention module in LSTM to learn which parts in the frames are fatal for the task at hand and assign higher importance to them. Spring from the soft-Attention LSTM, VideoLSTM (Li et al. 2018) hardwire convolutions and introduce motion-based attention to guides better the attention towards the relevant spatio-temporal locations. However, this complex architecture does not bring significant performance improvement. Recently, Sun et al. (Sun et al. 2017) propose Lattice-LSTM (L^2TSM), which extends traditional LSTM by learning independent hidden state transition of memory cells for individual spatial locations. But it does not address the complex backgrounds similar scenes problem in different categories very well. Additionally, Ma et al. (Ma et al. 2017) study two different ways, i.e., Temporal Segment LSTM and Temporal Inception, to extract spatio-temporal information by systematically explored possible network architectures. Wang et al. (Wang et al. 2016) propose a segmental network, which splits a long sequence into several segments followed by sparse sampling, and achieve state-of-the-art performances.

From a technical standpoint, our approach is based on the dynamic memory networks. Memory networks are typically used to tackle simple logical reasoning problem in natural language processing like question answering and sentiment analysis. The pioneering works Neural Turing Machine (NTM) (Graves, Wayne, and Danihelka 2014) and Memory Neural Networks (MemNN) (Sukhbaatar et al. 2015) both propose an addressable external memory with the reading and writing mechanism. For computer vision tasks, memory networks have been adopted in visual tracking (Yang and Chan 2018), video question answering (Gao et al. 2018) and so on. In recent work (Vu et al. 2018), an external memory module is employed to learn long-term online video representation in recurrent manner. Moreover, the proposed method is similar to attention-based action recognition model, which enhances the evident information by assigning soft weights on frames of one video. Except for temporal soft attention mechanism in soft-Attention LSTM (Sharma, Kiros, and Salakhutdinov 2015), Long et al. (Long et al. 2018b) propose an attention-based shifting operation to

integrate informative local features and an multimodal keyless attention model is proposed to fuse visual and acoustic features for action recognition in (Long et al. 2018a)

Approach

The primary goal of this paper is to enhance the model's capacity for learning long-term and complex motion for action recognition in videos, by capturing salient and informative motion dynamics across the long-range temporal structure. A structured and growing external memory has been demonstrated is capable of learning long-term sequential pattern in (Joulin and Mikolov 2015). Inspired by this idea, we propose to continuously read and write external memory module over time to obtain stable and discriminative representations of the human action captured by the video, which is verified by working with different CNN models.

Overview

We design a end-to-end dynamic memory network that writes task-relevant features into a growing external memory module according to a discrete memory controller. Figure 1 shows the overall architecture of our design. Specifically, given a whole video in the form of sparse sampled frame sequence with length T, suppose at each time t with $t = 1, \ldots, T$, the network first produces convolution features $\mathbf{x}_t$ with d dimension for individual frames via arbitrary convolutional network. As shown in many CNN-based method (Wang et al. 2016; Simonyan and Zisserman 2014), after training, these convolutional features can capture significant appearance evidences for action recognition. Subsequently, convolutional features $\mathbf{x}_{1,\ldots,t-1}$ are fed into a LSTM to obtain context embedding $\mathbf{ce}_t$ for each time t. Meanwhile, the memory history $\mathbf{mh}_t$ is calculated by current external memory module $\mathbf{M}_t$. At the end, the memory controller takes context embedding $\mathbf{ce}_t$, memory history $\mathbf{mh}_t$ and convolutional feature $\mathbf{x}_t$ as inputs and outputs a discrete decision $\mathbf{s}_t \in \{0, 1\}$ on whether write convolutional feature $\mathbf{x}_t$ into external memory module $\mathbf{M}_t$. The combination of context embedding and memory history enables the model to capture informative motion dynamics in long-range temporal structure.

The whole input sequence is processed in a sequential manner. The memory history $\mathbf{mh}_{t+1}$ is obtained after external memory module $\mathbf{M}_t$ have been updated at time t, and the features in external memory module $\mathbf{M}_T$ are averaged to get final representation for action recognition. In what follows, we will explain the read and write process of external memory over time with details, as well as how the discrete decisions are made by the memory controller.

Updating Memory Module over Time

Different from previous works that either mixes memory with computation in the recurrent network or mimics the one-dimensional memory with elaborate access mechanism in the Turing machine/von Neumann architecture, we propose a growing memory module to process arbitrarily long sequences theoretically. This is intuitive because frame sequences that capture same human action may have different length with different sample rate. But more importantly, a growing memory module is suitable for model long-term and complex motion in videos. To be specific, our memory module is represented by a list M, with the length of it equals to the number of memory items and memory item is D dimension feature vector $\mathbf{m}$. Memory module can be updated after processing each frame, thus we use $\mathbf{M}_t^{'}$ to represent memory before updating for the current frame so far and $\mathbf{M}_t$ denotes memory module after processing current frame, that is, $\mathbf{M}_t^{'}$ and $\mathbf{M}_{t-1}$ are the same.

Memory Read and Write. When processing the first frame, the external memory module is empty so that memory history $\mathbf{mh}_1$ is a zero feature vector. Suppose memory module $\mathbf{M}_t, t > 1$ is not empty with length $N_t > 0$, and then $\mathbf{mh}_t$ is average of all memory items in memory module,

$$\mathbf{mh}_t = \sum_{i=1}^{N_t} \mathbf{m}_i, \tag{1}$$

which is a simple but effective memory read mechanism. Meanwhile, context embedding $\mathbf{ce}_t$ is calculated by a custom LSTM, which will be elaborated in next section. Next, the context embedding $\mathbf{ce}_t$, memory history $\mathbf{mh}_t$ and convolutional feature $\mathbf{x}_t$ are fed into the memory controller to obtain discrete decision $\mathbf{s}_t \in \{0, 1\}$, and the memory module is updated as follows,

$$\mathbf{M}_t = \begin{cases} \varphi(\mathbf{M}_t^{'}, W_w \mathbf{x}_t) & \mathbf{s}_t = 1 \\ \mathbf{M}_t^{'} & \mathbf{s}_t = 0 \end{cases}, \tag{2}$$

where $W_w \in R^{D \times d}$ is a matrix that transform convolutional feature $\mathbf{x}_t$ into memory item $\mathbf{m}_{N_t+1}$, and the write function $\varphi(\cdot)$ is implemented as list append operation. Obviously, the external memory module is updated only when $\mathbf{s}_t = 1$ and remain the same when $\mathbf{s}_t = 0$. With large-scale datasets, the model can learn to write relevant information into memory module and neglect noise from scratch, which is crucial to model complex motion dynamics in long-range temporal structure. Except memory history and convolutional feature, context embedding is another factor to determine the memory write decision, and we will introduce it in the next section.

Context Embedding. Due to LSTM's limited ability to address the non-stationary issue of long-term motion dynamics (Sun et al. 2017), we build a segmental recurrent cell on top of LSTM unit, which alleviates the non-stationary issue by split a long sequence into several short segments. At its core, there is a one-dimensional vector $\mathbf{v}_t \in \{0, 1\}$ which indicates whether continue updating the LSTM's state at current timestep or reinitializes them to zeros. Because long-term motion dynamics usually shows segmental characteristics, in practice, we just replace this indicator vector with the memory discrete decision $\mathbf{s}_t$, i.e., $\mathbf{v}_t = \mathbf{s}_t$, which reduces the computation and makes sense to some extent. Specifically, indicator vector $\mathbf{v}_t$ decide whether to transfer the hidden state and memory cell content to the next timestep or to reinitialize them,

$$\hat{\mathbf{h}}_t, \hat{\mathbf{c}}_t = LSTM(\mathbf{x}_t; \mathbf{h}_{t-1}, \mathbf{c}_{t-1}), \tag{3}$$

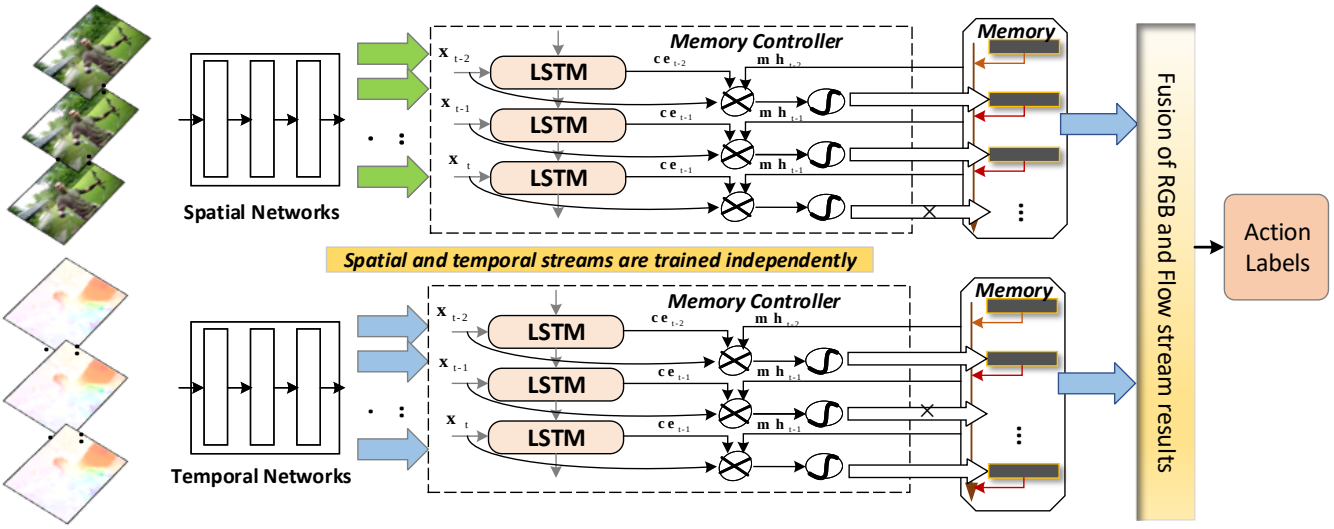

Figure 1: The overview of Memory-Augmented Temporal Dynamic Learning Network for action recognition. First, a set of video sequences are passing through CNN to extract convolutional features. Then, an LSTM unit is used to obtain context embedding by processing these features sequentially. Meanwhile, the memory items are recalled to constructed memory history. At each timestep, the memory controller takes these as inputs and outputs the discrete memory write decisions. Finally, the memory items in the memory module are pooled to recognizing action in the video. The final classification results are the average of spatial and temporal stream results.

$$\mathbf{h}_t = \mathbf{s}_t \times \hat{\mathbf{h}}_t, \tag{4}$$

$$\mathbf{c}_t = \mathbf{s}_t \times \hat{\mathbf{c}}_t. \tag{5}$$

The resulting state and memory are employed to compute gates values at the next time step. The LSTM breaks the connection with the previous hidden states $\mathbf{h}_{t-1}$, $\mathbf{c}_{t-1}$ and reinitializes them to zeros if $\mathbf{s}_t = 0$, and the hidden state of timestep t is passed to $t+1$ when $\mathbf{s}_t = 1$. We use $\hat{\mathbf{h}}_t$ as the context embedding $\mathbf{ce}_t$ at time t.

Discrete Memory Controller

In previous works on memory networks, the memory controller is constructed by a feed-forward network or LSTM, which interacts with the external memory module using a number of read and write instructions. We employ the similar idea but the controller only outputs the write instruction that acts to place informative features into memory module, because the read process is performed at each timestep. Specifically, for each time step, the memory controller takes the convolutional feature $\mathbf{x}_t$, memory history $\mathbf{mh}_t$ and context embedding $\mathbf{ce}_t$ as inputs, and outputs the discrete memory write decision $\mathbf{s}_t \in \{0, 1\}$. Formally, the write decision $\mathbf{s}_t$ is computed as a linear combination of these three inputs, followed by a function τ which is the composition of sigmoid function and a hard threshold function:

$$\mathbf{q}_t = \mathbf{v}_s^T \cdot ReLU(W_{sf}\mathbf{x}_t + W_{sc}\mathbf{ce}_t + W_{sm}\mathbf{mh}_t + \mathbf{b}_s), \tag{6}$$

$$\mathbf{a}_t = \sigma(\mathbf{q}_t), \tag{7}$$

$$\mathbf{s}_t = \tau(\mathbf{a}_t), \tag{8}$$

$$\tau(x) = \begin{cases} 1, & if\ x > thr \\ 0, & otherwise \end{cases}, \tag{9}$$

where $\mathbf{v}_s^T$ is learnable row vector and W_{sf}, W_{sc}, W_{sm} and $\mathbf{b}_s$ are learned weights and biases. Firstly, sigmoid function σ is applied on the outputs from linear combination to normalize $\mathbf{a}_t \in (0, 1)$, which represents the importance of current frame for recognizing action in the video. Then, we adopt a simple threshold-based selection mechanism to decide whether write current convolutional feature $\mathbf{x}_t$ into external memory module, where thr is a hyper-parameter and set to 0.5 in our experiments.

Straight-Through Estimator. The discrete decisions involved in transferring the binary variable $\mathbf{s}_t$ with the hard threshold function τ make it impossible to use standard gradient back-propagation to learn the parameters of the memory controller. To solve this issue, straight-through estimator proposed by (Bengio, Léonard, and Courville 2013) is employed to obtain to gradients of the memory controller. The idea is that a differentiable approximation function is used to substitute hard threshold function. In our case, we approximate the hard threshold function with the identity function, which has shown its effectiveness when considering a single layer of neurons. After replacing hard threshold function with identity function, the chain rule between $\mathbf{s}_t$ and $\mathbf{q}_t$ is

$$\begin{aligned} \frac{\partial \mathbf{s}_t}{\partial \mathbf{q}_t} &= \frac{\partial \mathbf{s}_t}{\partial \mathbf{a}_t} \cdot \frac{\partial \mathbf{a}_t}{\partial \mathbf{q}_t} = 1 \cdot \frac{\partial \mathbf{a}_t}{\partial \mathbf{q}_t} \\ &= \sigma(\mathbf{q}_t)(1 - \sigma(\mathbf{q}_t)), \end{aligned} \tag{10}$$

which enable the model can be trained in end-to-end manner. At inference time, the hard threshold function is used to output memory write signal. In this way, the discrepancy between training and test stage is eliminated in forward pass.

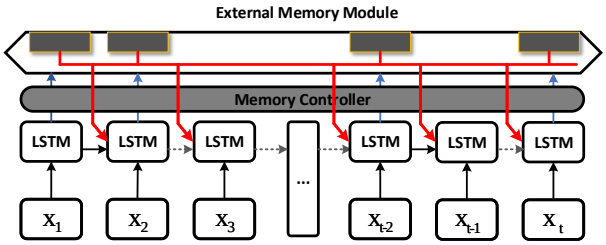

Figure 2: Illustration of the data flow in our proposed model. 'LSTM' boxes represent the linear operations in Eq. 3, the black solid line between 'LSTM' boxes represents the hidden states update operation and gray dashed line means reinitialize hidden state to zero, as shown in Eq. 4, Eq. 5. Additionally, blue line cross memory controller box means that corresponding features are written into memory module and the red line represents the read processing of external memory module.

Learning Architecture

As illustrated in Figure 1, the two-stream framework is adopted to improve the accuracy of prediction, where the optical flow is utilized as an additional modality to compensate the RGB inputs. The optical flow inputs are complementary to our proposed model, because optical flow only captures small motion between consecutive frames and our model aims to learn long-term motion patterns in long-range temporal structure. Note that video sequences, each composed of several frames sampled from a video, are fed into convolutional network to extract high-level feature representations. Therefore, our approach can be extendible for almost all CNN architectures, including BN-Inception, ResNets, and VGGnet. In practice, we initialize our convolutional network with models that are well pre-trained on ImageNet before we fine-tune them on the relatively small video datasets.

Computation Graph Analysis. From data flow perspective, our approach enables the model to reroute the forward path adaptively, which provides an effective and efficient way to exploit the temporal correlations in long-range motion dynamics. As shown in Figure 2, suppose the convolutional feature $\mathbf{x_t}$ is written into external memory module, i.e., $\mathbf{s_t} = \mathbf{1}$, and it will be connected with subsequent frames through memory read processing such as $\mathbf{x_1}$ and $\mathbf{x_2}$ in Figure 2. Meanwhile, the inherent connection in LSTM unit is broken using Eq. 4 and Eq. 5 when $\mathbf{s_t} = \mathbf{0}$, e.g. dashed line between $\mathbf{x_2}$ and $\mathbf{x_3}$ in the figure. Compared to traditional LSTM, all information from foregoing frames that are transformed into memory items are utilized to compute the gate value $\mathbf{s_t}$ rather than only last frame, which improves capacity for modeling long-term and complex motion dynamics. For example, the information of $\mathbf{x_1}$ and $\mathbf{x_2}$ is passed to frame $t-1$ and t through "red" data path in Figure 2. Moreover, all parameters in the memory controller are learned from video datasets and the data flow of forward pass varies in different video sequences by using sample-dependent variable $\mathbf{s_t}$.

Experiments

To evaluate the proposed network architecture, we conduct action recognition experiments on two benchmark datasets, with in-depth comparisons with baseline and other architectures to verify our design principles. We also provide visualization of the learned memory controller as a critic part of the model's interpretability.

Experimental Settings

Dataset. Experiments are mainly conducted on two action recognition benchmark datasets: UCF101 (Soomro, Zamir, and Shah 2012) and HMDB51 (Kuehne et al. 2011). The UCF101 dataset is composed of 13,320 videos categorized into 101 actions, ranging from daily life activities to unusual sports. The videos are collected from Internet with various camera motions and illuminations. HMDB51 dataset contains 6766 videos divided into 51 action classes. Complex backgrounds and similar scenes in different categories make this dataset more challenging than others. For both of them, we follow the provided evaluation protocol and adopt standard training/test splits and report the mean classification accuracy over these splits.

Implementation Details. We choose VGG16 and ResNet101 as our CNN feature extractor for the RGB and optical flow images. We use stochastic gradient descent algorithm to train our model from scratch. The mini-batch size is set to 64 and the momentum is set to 0.9. We use small learning rate in our experiments. For spatial-stream networks, the learning rate is initialized as 0.001 and decrease by $\frac{1}{10}$ every 6,000 iterations. The training procedure stops after 18,000 iterations. For the temporal stream, we initialize the learning rate as 0.005, which reduces to its $\frac{1}{10}$ after 48,000 and 72,000 iterations. The maximum iteration is set as 80,000. We use gradient clipping of 20 to avoid exploding gradient at the early stage. For data augmentation, we use the techniques of location jittering, horizontal flipping, corner cropping, and scale jittering, as described in (Wang et al. 2016). All the experiments are run on the PyTorch toolbox (Paszke et al. 2017).

Baseline Two-stream ConvNets. There are two types of two-stream baseline. One is the convNet-baseline reported in (Li et al. 2018) and another one is TSN-baseline proposed in (Wang et al. 2016). We distinguish these two baseline-based method by w/o "+TSN", i.e., Ours(VGG)/Ours(VGG+TSN) and Ours(ResNet+TSN).

Table 1: Performance evaluation on videos of split 1 of UCF-101 with complex movements.

Action Category	ConvLSTM	Ours(VGG)
Human-Object Interaction	76.0	83.3
Human-Human Interaction	89.1	91.5
Body Motion Only	88.1	89.3
Pizza Tossing	21.1	54.6
Mixing Batter	55.6	75.6
Playing Dhol	81.6	95.9
Salsa Spins	86.0	93.1
Ice Dancing	97.8	98.0

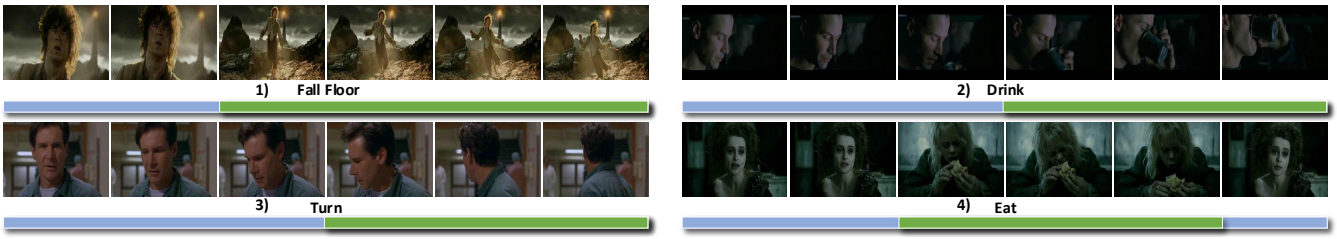

Figure 3: Memory controller decision results of the proposed method on HMDB51 dataset. Note that the memory controller can correctly select the most evident frames and ignore irrelevant frames. Specifically, the blue bar stands for the memory controller decision $\mathbf{s_t} = \mathbf{0}$, and the green bar stands for the memory controller decision $\mathbf{s_t} = \mathbf{1}$, i.e., corresponding convolutional features are written into external memory module.

Effect on videos with complex movements

In this subsection, We focus on the effectiveness of our model processing videos with complex motion dynamics. The 101 action categories of UCF101 dataset have been divided into five coarse types: human-object interaction, body-motion only, human-human interaction, playing musical instruments, and sports. We summarize the classification accuracy on these coarse categories and report it in Table 1. The upper part of the table shows the performance on the coarse category. The ConvLSTM that aims to model spatio-temporal pattern in complex motion dynamics are utilized for comparison, and our model outperforms on all coarse categories, particularly by 7.3% on human-object interaction, by 2.4% on human-human interaction and 1.2% on body motion only. The impressive improvements on human interacting actions verify our model's superiority in handling complex motion dynamics over ConvLSTM. Meanwhile, our method shows similar results with ConvLSTM on body motion only actions since the movements are simple in this category. Furthermore, results from several typical classes are shown in lower part of the table. Our model gains 33.5% and 20.0% improvements on pizza tossing and mixing batter category respectively, where the movements between the person and object are complex and fast. In the ice dancing category, we see only 0.2% improvement since the appearance information is very discriminative and recognition accuracy is comparable using one-single frame.

Table 2: Detailed results on videos of split 1 of UCF-101 using VGG16.

Method	Spatial Networks	Temporal Networks
ConvNet	77.4	75.2
LSTM	77.5	78.3
ALSTM	77.0	79.5
ConvLSTM	77.6	79.1
VideoLSTM	79.6	82.1
Ours(VGG)	80.0	82.7

Memory Controller Visualization

We present several video sequences to visually verify the effect of the discrete memory controller, which is the key component that rules the memory content for action recognition. Figure 3 shows the memory write decisions and corresponding RGB frame on split 1 of HMDB51 datasets. We can clearly see that the memory controller learns to write the most evident information into external memory module and ignore irrelevant frames.

In Figure 3, we show four video sequences sampled from fall floor, drink, eat, and turn categories. These actions have strong segmental characteristics, for example, drink can be divided into three stages: grab the glass, drink and put down the glass. And the learned memory controller can select the most discriminative stage to recognize actions. For instance, in sequence 1), there is a close-up of the person's head in the beginning which is irrelevant to the fall floor activity, so the memory controller decides to ignore these close-up frames and store the relevant frames once they appear. What is more, in sequence 2) and 3), drink and turn activities are composed of several stages and the activities are usually confusing before real actions occur. For example, the person may put down the glass instead of drink it after observed three frames in sequence 2). Therefore, the learned memory controller tends to write the most evident frames into the external memory module even though losing some relevant information. At the same time, the learned memory controller is robust to background and object variation. The changes of person and background make the sequence 4) more challenging, and our controller can autonomically find the leading person and ignore others.

Comparison with other temporal models

Recurrent neural network, especially LSTM, have the ability to model temporal motion dynamics, and shown moderate improvement from previous work. Moreover, there are several LSTM variants aiming at learning complex motion dynamics from video, such as ConvLSTM, soft-Attention LSTM (ALSTM) and VideoLSTM (Li et al. 2018). Due to the VGG16 convolutional network are used as feature extractor in these methods, we choose VGG16 network as base architecture and list detailed results in Table 2. The vanilla LSTM only obtains no improvement on spatial network and 3.1% improvement on temporal network. By enhancing the network with the external memory module, our model produces the highest accuracy in both stream network and im-

Table 3: State-of-the-art comparison with LSTM-like architectures.

Method	Pre-training		Fusion		Networks		UCF101	HMDB51
	ImageNet	1M Sports	Average	Product	VGG	ResNet		
ConvNet	✓	-	✓	-	✓	-	77.4	75.2
ALSTM	✓	-	✓	-	✓	-	77.0	41.3
VideoLSTM	✓	-	-	✓	✓	-	89.2	56.4
L^2STM	✓	-	✓	-	✓	-	93.6	66.2
TSN(ResNet)	✓	-	✓	-	-	✓	93.9	69.7
TS-LSTM	✓	-	✓	-	-	✓	94.1	69.0
Ours(VGG+TSN)	✓	-	✓	-	✓	-	92.1	67.3
Ours(ResNet+TSN)	✓	-	✓	-	-	✓	94.8	71.8

proves the performance by 2.6 and 7.5 points respectively. Furthermore, our model also outperforms other LSTM variants that are designed to action recognition.

In Table 3, we list all state-of-the-art methods which learn motion patterns with LSTM for action recognition tasks. In order to present the completely and clearly, we elaborate on some factors, such as pre-training type, fusion strategy, and network architectures. To be specific, we compare the proposed method with ALSTM (Sharma, Kiros, and Salakhutdinov 2015), VideoLSTM (Li et al. 2018), L^2STM (Sun et al. 2017) and TS-LSTM (Ma et al. 2017). The recent state-of-the-art method L^2STM and TS-LSTM also aim at modeling long-term motion dynamics in the videos, and our method outperforms L^2STM in terms of average accuracy 67.3% vs 66.2% on HMDB51 dataset and obtains comparable performance on UCF101 dataset. The reason behind this is the videos from UCF101 are trimmed carefully, which makes our method less useful, and a cross-modal training strategy is utilized to boost final performance. Compared with UCF101 dataset, videos from HMDB51 dataset usually contains some action-irrelevant shot, as shown in Figure 3, which makes HMDB51 dataset more challenging. And our proposed model tackles these video better by storing useful information and ignore noisy feature representations, so our approach makes a consistent improvement on HMDB51 dataset when compared with other methods.

Comparison with the state-of-the-art

In addition to the various temporal models comparison in Table 3, we compare our method with other state-of-the-art algorithms and the results are reported in Table 4. We evaluate our method following the testing scheme described in the standard two-stream method (Simonyan and Zisserman 2014), where final classification results are obtained by average of the spatial and temporal stream results. Specifically, we compare our method with traditional approaches such as improved trajectories (iDTs) (Wang et al. 2011) and deep learning algorithms such as two-stream networks (Simonyan and Zisserman 2014), factorized spatio-temporal convolutional networks (F_{ST}CN) (Sun et al. 2015), 3D convolutional networks (C3D) (Tran et al. 2015), spatio-temporal fusion CNNs (Feichtenhofer, Pinz, and Zisserman 2016), attention cluster (Att-C) (Long et al. 2018b) and TSN (Wang et al. 2016), which model long-term temporal structure by segmental architecture with sparse sampling.

As shown in Table 4, our best implementation based on ResNet improves the average accuracy by 0.7% on UCF101 and 2.5% on HMDB51, which is reported in TS-LSTM using the same ResNet architecture. For testifying that our method is generally effective, we additionally evaluate our model with VGG-16 network as the previous two-stream CNNs architectures (Simonyan and Zisserman 2014). Both based on VGG-16, our method (92.1%) is still comparable to the spatio-temporal fusion CNNs (92.5%) (Feichtenhofer, Pinz, and Zisserman 2016) on UCF101 datasets and outperforms it by 1.9% on HMDB51 dataset. Meanwhile, our method achieves the best results 94.8% and 71.8% on UCF101 and HMDB51 dataset respectively. This result demonstrates that our approaches can be widely applied to many fancy CNN models.

Table 4: Performance comparison with the state-of-the-art.

Model	Method	UCF101	HMDB51
Traditional	iDT + FV	85.9	57.2
	iDT + HSV	87.9	61.1
	VideoDarwin	-	63.7
	MPR	-	65.5
Deep	Two Stream	88.0	59.4
	F_{ST}CN	88.1	59.1
	VideoLSTM	89.2	56.4
Very Deep	C3D	85.2	-
	Fusion	92.5	65.4
	L^2STM	93.6	66.2
	TS-LSTM	94.1	69.0
	TSN(BN-Inception)	94.0	68.5
	TSN(ResNet)	93.9	69.7
	Att-C	94.6	69.2
Ours	Ours(VGG16+TSN)	92.1	67.3
	Ours(ResNet+TSN)	**94.8**	**71.8**

Conclusion

In this paper, we present a memory-augmented temporal dynamic learning network to address the challenging problem of modeling long-term and complex motion dynamics. Particularly, we attach a simple and growing external memory module to CNN to store the most evident and relevant information for recognizing action in the video. Fur-

thermore, through memory items in the external memory module, the model can route the forward path adaptively using the discrete memory controller. As the core of the network, the memory controller employs hard threshold function to output discrete memory write decision and approximates its gradients using straight-through estimator, which enables the network can be trained end-to-end from scratch. As shown in visualization results, the learned memory controller can write the most evident information into external memory module and ignore irrelevant frames automatically. Moreover, to verify the generalization ability of the proposed model, we evaluate our model with different CNN architecture. Experiments conducted on both UCF101 and HMDB51 datasets validate our proposal and analysis.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2017YFB1002202, National Natural Science Foundation of China under Grant U1864204 and 61773316, Natural Science Foundation of Shaanxi Province under Grant 2018KJXX-024, Projects of Special Zone for National Defense Science and Technology Innovation, Fundamental Research Funds for the Central Universities under Grant 3102017AX010, and Open Research Fund of Key Laboratory of Spectral Imaging Technology, Chinese Academy of Sciences.

References

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Boiman, O., and Irani, M. 2007. Detecting irregularities in images and in video. *International journal of computer vision* 74(1):17–31.

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

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2017. Spatiotemporal multiplier networks for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2097–2106.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1933–1941.

Gao, J.; Ge, R.; Chen, K.; and Nevatia, R. 2018. Motion-appearance co-memory networks for video question answering. *arXiv preprint arXiv:1803.10906*.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Joulin, A., and Mikolov, T. 2015. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Advances in Neural Information Processing Systems*, 190–198.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. Hmdb: a large video database for human motion recognition. In *IEEE International Conference on Computer Vision*, 2556–2563.

Li, Z.; Gavrilyuk, K.; Gavves, E.; Jain, M.; and Snoek, C. G. 2018. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding* 166:41–50.

Lin, W.; Sun, M.-T.; Poovandran, R.; and Zhang, Z. 2008. Human activity recognition for video surveillance. In *International Symposium on Circuits and Systems*, 2737–2740.

Long, X.; Gan, C.; de Melo, G.; Liu, X.; Li, Y.; Li, F.; and Wen, S. 2018a. Multimodal keyless attention fusion for video classification. In *AAAI Conference on Artificial Intelligence*.

Long, X.; Gan, C.; de Melo, G.; Wu, J.; Liu, X.; and Wen, S. 2018b. Attention clusters: Purely attention based local feature integration for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 7834–7843.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Ma, C.-Y.; Chen, M.-H.; Kira, Z.; and AlRegib, G. 2017. Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *arXiv preprint arXiv:1703.10667*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshop*.

Sharma, S.; Kiros, R.; and Salakhutdinov, R. 2015. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 568–576.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, 2440–2448.

Sun, L.; Jia, K.; Yeung, D.-Y.; and Shi, B. E. 2015. Human action recognition using factorized spatio-temporal convolutional networks. In *IEEE International Conference on Computer Vision*, 4597–4605.

Sun, L.; Jia, K.; Chen, K.; Yeung, D.-Y.; Shi, B. E.; and

Savarese, S. 2017. Lattice long short-term memory for human action recognition. In *IEEE International Conference on Computer Vision*, 2166–2175.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, 4489–4497.

Vu, T.-H.; Choi, W.; Schulter, S.; and Chandraker, M. 2018. Memory warps for learning long-term online video representations. *arXiv preprint arXiv:1803.10861*.

Wang, H.; Kläser, A.; Schmid, C.; and Liu, C.-L. 2011. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3169–3176.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, 20–36.

Wang, Q.; Yuan, Z.; Du, Q.; and Li, X. 2018. Getnet: A general end-to-end 2-d cnn framework for hyperspectral image change detection. *IEEE Transactions on Geoscience and Remote Sensing*.

Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, 802–810.

Yang, T., and Chan, A. B. 2018. Learning dynamic memory networks for object tracking. *arXiv preprint arXiv:1803.07268*.

Yue-Hei Ng, J.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; and Toderici, G. 2015. Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4694–4702.

Zha, S.; Luisier, F.; Andrews, W.; Srivastava, N.; and Salakhutdinov, R. 2015. Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*.