MFI: Multi-range Feature Interchange for Video Action Recognition

Sikai Bai, Qi Wang* and Xuelong Li
School of Computer Science and Center for OPTical IMagery Analysis and Learning
Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China
E-mail: whitesk19@mail.nwpu.edu.cn, crabwq@gmail.com, xuelong_li@nwpu.edu.cn

***Abstract*—Short-range motion features and long-range dependencies are two complementary and vital cues for action recognition in videos, but it remains unclear how to efficiently and effectively extract these two features. In this paper, we propose a novel network to capture these two features in a unified 2D framework. Specifically, we first construct a Short-range Temporal Interchange (STI) block, which contains a Channels-wise Temporal Interchange (CTI) module for encoding short-range motion features. Then a Graph-based Regional Interchange (GRI) module is built to present long-range dependencies using graph convolution. Finally, we replace original bottleneck blocks in the ResNet with STI blocks and insert several GRI modules between STI blocks, to form a Multi-range Feature Interchange (MFI) Network. Practically, extensive experiments are conducted on three action recognition datasets (i.e., Something-Something V1, HMDB51, and UCF101), which demonstrate that the proposed MFI network achieves impressive results with very limited computing cost.**

I. Introduction

Video action recognition is a fundamental yet challenging task in the field of computer vision. It involves recognizing the human actions in videos and has gained much attention from academia and industry over recent years. Different from the image, video is a sequence of frames with complex temporal evolution. Temporal modeling for video action recognition is usually considered in multiple ranges, including short-range motion encoding between adjacent frames and long-range dependency learning at the large temporal range. As shown in Fig 1(a), action instance on a single frame is related to the objects and background both in short-range and long-range.

In recent years, many methods have been proposed to consider one or both of these ranges, but it is still unclear how to capture temporal information with complex evolution on multiple ranges using an efficient and effective way. Two-stream CNNs [1], [2], [3] focus on learning the discriminative temporal features from the optical flow. Although two-stream CNNs have made performance improved, the optical flow only represents the motion features between the neighboring frames and extracting optical flow is usually expensive in both space and time. LSTM regards the video as an ordered sequence of frames to capture the temporal relationship between these frames, but LSTM-based methods [4], [5], [6] normally cannot model the complex temporal relationship among frames well. Recently, by stacking 3D convolution, 3D convolutional networks (3D CNNs) based methods [7], [8], [9] are proposed to capture short-range temporal features and long-range dependencies directly from RGB input frames. 3D CNNs based methods are capable of achieving outstanding performance, but they suffer from heavy computational cost, which limits the real-world deployments.

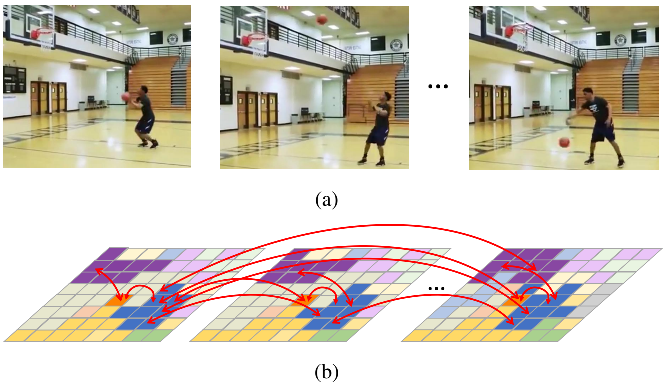

Fig. 1. The illustration of our main idea. (a) Original video. Human action on a single frame can be effectively determined by referring to the objects and background both in the short-range and long-range; (b) The features from the colored regions bi-directionally shift in the feature map of video models.

To remedy the aforementioned problems, we propose a novel and effective method, referred as MFI network, which regards short-range motion encoding and long-range dependency learning as the interchange between features in multiple ranges. As shown in Fig 1(b), we construct a feature interchange operation by bi-directionally shifting features in the feature map. Specifically, for short-range motion encoding, we propose a channels-wise temporal interchange (CTI) module. CTI module first gains temporal difference related to motion information and then interchanges temporal difference along the temporal dimension with both previous and subsequent frames. Finally, we insert an identity mapping path to combine the original features with interchanged features. CTI module makes the current frame obtain interchanged information and retain the original features. For long-range dependency learning, we devise a Graph-based Regional Interchange (GRI) module. GRI module first transforms the features in a regular feature map into the state of nodes in a non-grid graph, and then realize long-range features interchange and fusion with graph convolution. Eventually, we reverse the state of nodes

Qi Wang is the corresponding author.

into the features in a regular feature map to be compatible with CNN models.

To construct a multi-range feature interchange (MFI) network, we use ResNet as the backbone. The proposed CTI module is inserted into an original bottleneck block in the backbone to form a short-range temporal interchange (STI) block. The entire network is built by replacing the original bottleneck blocks with STI blocks and insert several GRI modules between STI blocks. Benefiting from the proposed two modules are not only complementary, but they only introduce very limited additional computing costs, the proposed MFI network is efficient and effective. Furthermore, comprehensive experiments on Something-Something V1, UCF101, and HMDB51 demonstrate our MFI network gains comparable performance to the state-of-the-art methods with the help of the very limited computing cost.

Overall, our main contribution can be summarized as follow:

- We devise a channel-wise temporal interchange (CTI) module, which is constructed by performing channel-wise temporal interchange along the temporal dimension to effectively encode short-range motion features.
- We build a graph-based regional interchange (GRI) module, which learns efficiently long-range dependencies by interchanging distant regional features using graph convolution and can be compatible with CNN models.
- We propose a novel multi-range feature interchange (MFI) network, which integrates the proposed two modules to perform temporal modeling in short-range and long-range. Extensive experiments on three benchmark datasets demonstrate our MFI network offers comparable performance to the state-of-the-art methods using very limited computing cost.

II. Related Work

A. Action Recognition

There are extensive studies in action recognition. Early approaches rely on extracting hand-crafted features to learn video representation [10], [11], [12]. And then with the great success of deep learning methods in the computer vision area, many researchers attempted to apply deep networks to video action recognition. Among them, Simonyan *et al.* [2] proposed a two-stream network, which takes the RGB frames and the stacked optical flow frames as input for extracting spatial features and temporal information in two CNN branches, respectively. Wang *et al.* [13] proposed Temporal Segment Network to perform the sparse sampling strategy for long-range video clips and learn temporal evolution between the sampled frames. Additional 2D CNNs methods include Convolution Fusion [14], Temporal Relation Network [15], and TSIN [16]. These approaches are sufficiently straightforward and effective, but they either require additional optical flow modality or cannot capture complex temporal relationships well. Moreover, some 2D CNN+LSTM works [4], [17], [5], [6] have been proposed, which regard the video as an ordered sequence of frames, and capture the temporal relationship by using LSTM to aggregate 2D CNN features extracted from individual frames. In those methods, since the feature learning of each frame is isolated and only the high-level 2D CNN features are utilized for temporal relationship modeling, they normally cannot capture the complex temporal relationship among frames well. Recently, some researchers introduce 3D CNNs. Tran *et al.* [18] proposed C3D architecture to learn spatiotemporal information jointly. Carreira *et al.* [7] inflated 2D convolutional kernels into 3D on an Inception V1 model [19]. Hara *et al.* [8] attempted inflated 2D convolutional kernels into 3D on ResNet and some derivate models of ResNet. After trained on large-scale datasets, these inflated 3D models obtained significant improvements in performance. However, 3D CNNs have quadratic growth of parameter and high computational costs compared to 2D counterparts, making them more prone to overfitting.

B. Trade-off between Performance and Efficiency

Many attempts have been made to trade off performance and efficiency. Lee *et al.* [20] constructed a motion filter to attain spatio-temporal features from 2D CNN. Qiu *et al.* [21] and Sun *et al.* [22] decomposed 3D convolution into 2D convolution follow by 1D convolution. Xie *et al.* [23] built mixed convolutional models, where 3D convolution was used in either the top or bottom layers and 2D convolution in the rest. Top-heavy architecture has also been attempted in ECO [24]. In general, the above approaches alleviated the rapid growth of computational cost by 3D convolution. But most of them still not have the same order of magnitude computational cost as 2D competitors. The most recent work TSM [25] is designed to shift the original features on part of the channels along the temporal dimension, but this method neglects long-range dependencies. In our work, the proposed CTI module interchanges the temporal difference related to motion information between adjacent frames and preserves the original spatiotemporal information by an identity mapping connection. We also consider the long-range dependencies by the proposed GRI module.

C. Long-range Dependency

There are some works [26], [27],[28], [29] to capture long-range dependencies in videos with graph structure. Among them, Wang *et al.* [27] proposed to use space-time region graphs to represent videos and followed by graph convolutions for inferring relationships between objects, in which the objects need to be detected using an object detector trained on extra annotated data. Zeng *et al.* [28] proposed to construct graph convolutional networks for capturing temporally disjoint information. And Xu *et al.* [29] regards video snippets as nodes in a graph and form edges between them based on both their temporal ordering and semantic similarity. Attention mechanism is also proved to be effective for long-range dependency learning [30], [31], [32], [33]. For example, Non-local Network [33] attempts to deliver temporal dependencies from one place to another. Moreover, Temporal 3D ConvNets [34], DynamoNet [9] and LGD [35] are effective for

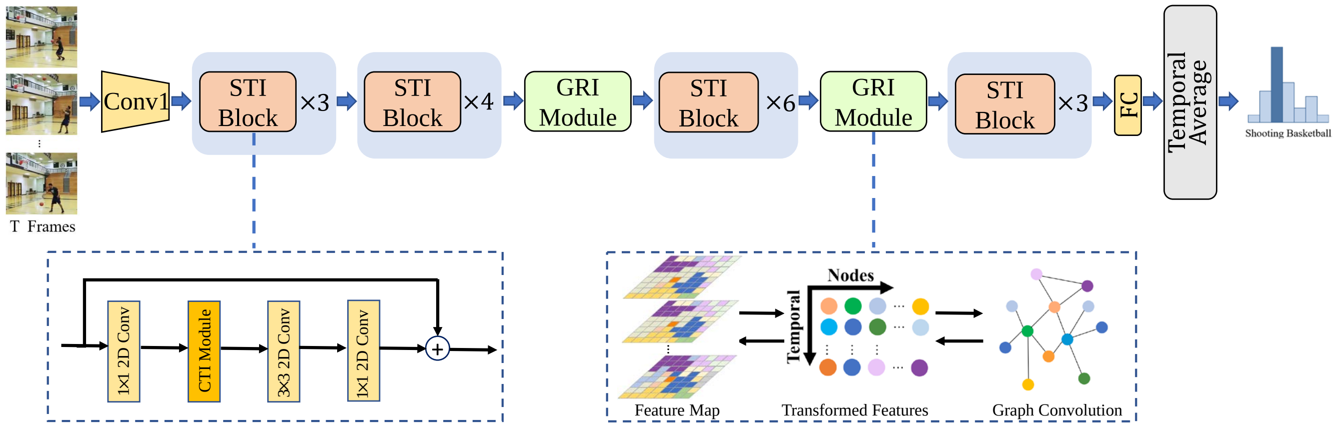

Fig. 2. The overview architecture of Multi-range Feature Interchange Network for video action recognition. Following the sparse sampling strategy [13], We adopt T sampled frames obtained from a video as the input of the network. 2D ResNet-50 is utilized as the backbone and all original bottleneck blocks are replaced by the proposed STI blocks ($\times$ n represents the number of STI blocks is n in a stage), we also insert two GRI modules between middle and top STI blocks in the network architecture. The global temporal pooling is applied to average action predictions for all of the sampled frames.

learning long-range dependencies. Nevertheless, most of these approaches are either computationally heavy or require extra data annotations. Different from previous work, our proposed GRI module projects the features from the grid map into the state of nodes in a non-grid and performs information interchange between distant features using graph convolution. There are no extra object detectors, data annotations and 3D convolution required for the entire process of our GRI module.

III. Approach

In this section, the proposed method will be described in detail. Specifically, we first give elaborative procedures of the proposed CTI module and GRI module and show how to capture short-range motion features and learn long-range dependencies, respectively. Afterward, We will present how to assemble CTI module and GRI module to form our multi-range feature interchange (MFI) network. The overview architecture of our MFI network is illustrated in Fig 2.

A. Channel-wise Temporal Interchange (CTI) Module

The existing approaches directly shift original features or extract optical flow to capture temporal features. Different from these works, the intuition behind the proposed CTI module is that, among all features, different features would focus on distinct information. A part of features tends to describe the static information related to background scenes; other features mainly focus on capturing the motion information about temporal evolution. For video action recognition, it is beneficial to enable the model to discover and then interchange those motion information.

The architecture of CTI module is shown in Fig 3. The goal of CTI module is to discover and then interchange the temporal difference related to motion information in the short temporal range. Given a spatiotemporal feature map $X \in R^{T\times H\times W\times C}$ as input, T represents the range of temporal evolution, H and W are the height and width of spatial representation and C is the number of channels. A 2D convolutional layer with kernel size 1×1 is first utilized to reduce the number of channels by a factor r for efficency. In our experiments, we set r to 16. Therefore, we obtain a compressed spatiotemporal feature map $Y \in R^{T\times H\times W\times C/r}$.

The temporal difference can be obtained by calculating the difference between the features of two consecutive frames (e.g. Y^t and Y^{t+1}). In fact, experimental evidence shows that different channels focus on different features, and the computation cost is able to be reduced by a factor of G where G represents the number of channel groups. Thus we set G to C/r, apply 2D channel-wise convolution on features Y^{t+1} and then subtract from Y^t to obtain temporal difference H^t in time t. The calculation process for each channel can be formally expressed as:

$$H_c^t = Conv_{trans} \otimes Y_c^{t+1} - Y_c^t, \quad t \in [1, T-1], \qquad (1)$$

here $Conv_{trans}$ represents 2D channel-wise convolution with kernel size 3×3. $\otimes$ and c indicate the convolutional operation and c-th convolutional channel, respectively. $c \in [1, C/r]$. Similarly, H^{t-1} can be acquired by performing 2D channel-wise convolution on features on Y^t and then subtracting from Y^{t-1}. To keep the temproal length compatible with the temproal evolution range of the input spatiotemporal feature map, we claim the temporal difference in time T as zero, i.e. $H^T = 0$.

Then we perform temporal difference interchange operation for the obtained temporal difference. Specifically, inspired by previous work [25], the proportion of interchanged channels is 1/4, where 1/8 of the channels are exchanged with the previous timestamp, and 1/8 of the channels are transferred with the subsequent one. The temporal interchange operation is formally described as:

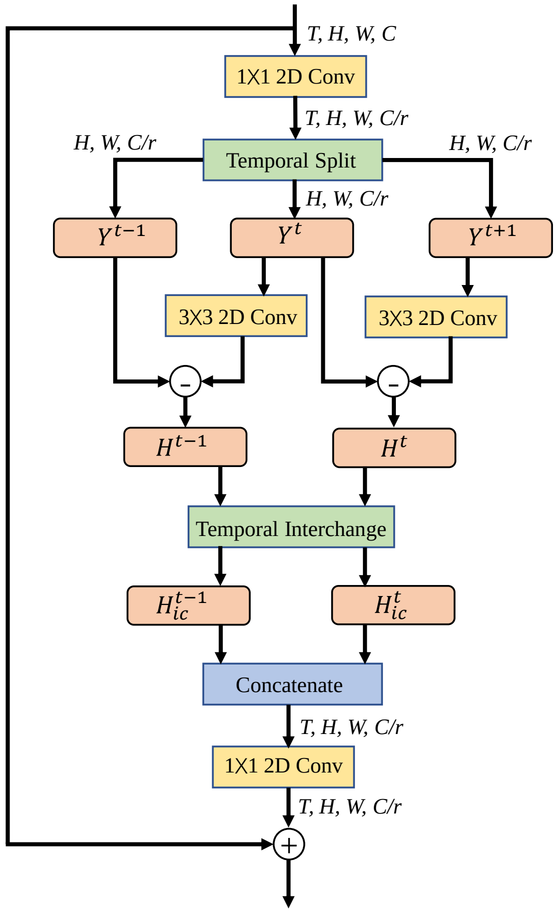

Fig. 3. The architecture of the channel-wise temporal interchange (CTI) module. The feature maps are represented as the shape of their tensors.

$$
\begin{aligned}
H_{ic}^{t}[h,w,c] &= H^{t+1}[h,w,c], \quad t \in [1, T-1],\ c \in [0, C/8r],\\
H_{ic}^{t}[h,w,c] &= H^{t-1}[h,w,c], \quad t \in [2, T],\ c \in [C/8r, C/4r],\\
H_{ic}^{t}[h,w,c] &= H^{t}[h,w,c], \quad t \in [1, T],\ c \in [C/4r, C/r],
\end{aligned} \tag{2}
$$

where H_{ic}^{t} is the obtained feature map by performing the temporal interchange operation for temporal difference, $t \in [1, T]$. We then concatenate all of the interchanged features along the temporal dimension and use another 2D convolutional layer with kernel size 1×1 to expand the number of channels to C. In order to avoid hurting the original spatiotemporal features, an identity mapping is built to combine the input spatiotemporal feature map with the output of CTI module.

Finally, we embed the proposed CTI module into the original bottleneck block in ResNet to form a new residual block, namely STI block. In a STI block, the first 2D convolutional layer with kernel size 1×1 is utilized to reduce the number of channels, and the proposed CTI module is used to discover and interchange temporal difference from compressed features. Afterward, a 3×3 2D convolutional layer and a following 1×1 2D convolutional layer are applied to extract spatial features and restore the number of channels. Overall, our STI block not only learns motion-related information but takes spatial structure modeling into consideration.

B. Graph-based Regional Interchange (GRI) Module

Previous video action recognition methods typically infer relationships between annotated objects or stack many 3D convolutional operations to learn the long-range dependencies. They are inefficient approaches since extra object detection, data annotation and a large number of 3D convolutional operations are computationally expensive. To remedy this issue, we build a GRI module to efficiently learn long-range dependencies. The proposed module consists of feature transformation, graph convolution, and feature reverse. We first construct a graph and transform the features in a regular feature map into the state of nodes over the graph. Then the state of nodes is interchanged and fused with each other using graph convolution. Finally, the output of graph convolution process is reversed into the features in a regular feature map. The detailed description of the proposed GRI module is shown below.

A graph $G(V, E)$ is designed to perform interaction between feature pairs. Specifically, $G(V, E)$ contains N nodes, where each node encodes the feature contained in the feature map as its state, and edges store the relationship between the underlying features of node pairs.

1) Feature Transformation: To implement transforming from the features in a regular feature map to the state of nodes in a non-grid graph G, let us hypothesize a given feature map $X \in R^{T \times H \times W \times C}$ in a video CNN model. We first obtain a reshaped feature map $X_{re} \in R^{L \times C}$ by reshaping X, where $L = T \times H \times W$. And then, we apply the 1D convolution and transpose operation for X_{re} to gain the transform weight matrix $W_t \in R^{N \times L}$, where $N = \lfloor C/4 \rfloor$. Finally, a transformed feature map V_t can be acquired by matrix multiplication between W_t and X_{re}, and we project the features in V_t as the state of nodes in the graph G. Formally, the entire feature transformation process can be formulated as:

$$
\begin{aligned}
W_t &= [Conv_{trans}^{'} \otimes \Phi_r(X)]^T, & W_t \in R^{N \times L},\\
V_t &= W_t * \Phi_r(X), & V_t \in R^{N \times C},
\end{aligned} \tag{3}
$$

here $Conv_{trans}^{'}$ represents a 1D convolution layer with kernel 1 for obtaining transformation weight; $\otimes$ denotes the convolutional operation; Φ_r is the reshape operation; T and $*$ indicate the transpose operation and matrix multiplication, respectively.

2) Graph Convolution: When applying graph convolution, each node propagates its state to the rest of the nodes and aggregates information from others over graph G. For example, two nodes contain regional features that focus on the player and the basketball respectively, which learn a connection and interchange information between each other in the long temporal range, it is helpful to recognize the human action in videos. Concretely, we use a single layer graph convolution network in [26] as our graph convolutional operation:

$$F(V_t, A_g, W_g) = (V_t - A_g V_t) W_g, \tag{4}$$

where $A_g \in R^{N \times N}$ is the node adjacency matrix, and $W_g \in R^{C \times C}$ is the state update parameter matrix. A_g and W_g are trainable and they can be randomly initialized and continually optimized during training. Then we aggregate the output of GCN and the input V_t as the output of the graph convolution process, which can be given by:

$$V_{out} = ReLU(F(V_t, A_g, W_g) + V_t), \tag{5}$$

here $ReLU$ is the rectified linear unit as the activation function.

3) Feature Reverse: After graph convolution, we reverse the output V_{out} into the features of a regular feature map to be compatible with 2D CNN models. This process is the inverse of the feature transformation process. Formally, the process of feature reverse can be denoted as follow:

$$Y_{inv} = \varphi_r(W_t^T * V_{out}), \tag{6}$$

where Y_{inv} represents the output of the feature reverse process. W_t^T indicates the transposed matrix for the transform weight matrix W_t. φ_r is the inverse of reshape operation Φ_r.

C. MFI Network

In order to perform multi-range feature interchange, we utilize 2D ResNet-50 as the backbone and construct a novel network to assemble the proposed CTI module and GRI module, namely MFI network. As we all know, 2D ResNet-50 is able to be divided into six stages, where stage 2 to stage 5 can be called conv2_x to conv5_x. Specifically, as shown in Fig 2, for short-range motion features, the CTI module is utilized to discover and interchange the temporal difference related to motion information. We embed the CTI module into the original bottleneck block in 2D ResNet architecture to form an STI block, and then we replace all original bottleneck blocks from conv2_x to conv5_x with STI blocks. For long-range dependencies, because empirical evidence shows feature maps in middle and top layers are more abundant in semantics and much smaller in size than feature maps in bottom layers for a CNN architecture, for reducing the number of parameters and computational costs, we just insert two GRI modules into the designed network, one between conv2_x and conv3_x and the other between conv3_x and conv4_x. Following the previous method [13], the global temporal average pooling is utilized at the last stage of the model to average action score of all of the sampled frames, and the average score can be regarded as the action prediction of the entire video.

IV. Experiments

In this section, we first give an introduction to experimental datasets and implementation details. We then report the performance of the proposed MFI network on different datasets and compare them with some state-of-the-art approaches. Finally, the ablation study is conducted. We testify the effectiveness of different components within the proposed MFI network and analyze the efficiency and performance of our MFI network.

A. Datasets and Implementation Details

1) Datasets: The proposed approach is evaluated on three benchmark datasets, Something-Something V1 [36], HMDB51 [37] and UCF101 [38]. Something-Something V1 contains more than 100,000 videos across 174 classes, collected for generic human-object interaction. It contains many video actions with ambiguous activity categories and related to temporal order. Such as 'Tearing Something into two pieces' versus 'Tearing Something just a little bit', and 'Moving something away from something' versus 'Moving something closer to something'. UCF101 dataset includes 13320 video clips collected from Youtube with various camera motions and illuminations, annotated into 101 action classes, which is composed of three training and test splits. HMDB51 dataset contains 6766 video clips categorized into 51 classes, the content of clips include daily life activities and unusual sports. There are also three training and test splits for HMDB51.

2) Implementation Details: Following the sparse temporal sampling strategy in TSN [13], we first evenly divided a given video into T segments. Then one frame is randomly selected from each segment to form the input sequence with T frames. Afterward, the size of the short side of each frame is fixed to 256, and corner cropping and random scaling are applied for data argumentation. We then resize these frames to 224x224 for network training. Therefore, the input size of the network is set as $T \times 224 \times 224$ consisting of T sampled frames with resolution 224×224. In our experiments, T is set to 8 or 16.

We choose 2D ResNet-50 as the backbone and SGD to train the network. The training parameters include momentum 0.9, and weight decay 0.001. We evaluate performance with accuracy and measure the efficiency with FLOPs, $i.e.$ floating-point multiplication-adds. For Something-Something V1, the parameters contain 35 epochs, batch size 16 and dropout 0.5. We initialize the learning rate to be 0.001 and decrease it by 10 every 15 epochs. And for UCF101 and HMDB51, the parameters include 25 epochs, batch size 16, and dropout 0.8. We use a small initial learning rate of 0.0005 and divide it by 10 every 10 epochs.

B. Benchmark Comparison

1) Something-Something V1: In this section, we first evaluate the performance of the MFI network by the comprehensive statistics (e.g. inference protocols, FLOPs, and recognition accuracy) on Something-Something V1. As shown in Table I, we list the comparison results of our MFI network with some state-of-the-art methods on Something-Something V1. Specifically, the methods in the first compartment are based on 2D CNNs. Compared with the baseline model TSN, the proposed MFI network gains 24.2% top-1 accuracy improvement with the same number of input frames, while the FLOPs of MFI network slightly increases to 33.6G (1.02×), which demonstrates the effectiveness and efficiency of multi-range feature interchange. When using 8 and 16 input frames respectively, our MFI network achieved 0.4% and 0.8% performance improvements compared to TSM.

TABLE I
THE COMPARISON OF PERFORMANCE ON SOMETHING-SOMETHING V1.

Method	Backbone	#Frames	FLOPs	Val-Top1 (%)	Val-Top5 (%)
TSN [13]	BNInception	8	16G	19.5	-
TSN [13]	ResNet-50	8	33G	19.7	46.6
MultiScale TRN [15]	BNInception	8	16G	34.4	-
TSM [25]	ResNet-50	8	33G	43.4	73.2
TSM [25]	ResNet-50	16	33G	44.8	74.5
ECO_{8f} [24]	BNInception+3D ResNet18	8	32G	39.6	-
ECO_{16f} [24]	BNInception+3D ResNet18	16	64G	41.4	-
I3D [7]	3D ResNet50	32 × 2	153G × 2	41.6	72.2
Non-Local-I3D [33]	3D ResNet50	32 × 2	168G × 2	44.4	76.0
MFI(Ours)	ResNet-50	8	33.6G	43.9	73.9
MFI(Ours)	ResNet-50	16	67.2G	45.5	76.0

The second compartment contains the methods based on 2D+3D and 3D CNNs. Among them, the most efficient method is ECO, which is based on 2D+3D CNN and the FLOPs are only 32G. Compared with ECO, the FLOPs of our MFI network slightly increase 1.05×, but the performance is increased by a big margin, a relative improvement of 4.3% top-1 accuracy (39.6% vs. 43.9%) using 8 input frames and 4.1% top-1 accuracy (41.4% vs. 45.5%) using 16 ones. For 3D CNN based methods, because of the heavy computing costs, the FLOPs of these methods are significantly higher than our MFI network, but the proposed MFI network with 16 input frames still outperforms than I3D and Non-Local I3D by 5.5% and 1.1% on top-1 accuracy, respectively. These results demonstrate the effectiveness of our MFI network to learn multi-range temporal information by feature interchange operation on Something-Something V1 is quite impressive.

2) UCF101 and HMDB51: We then evaluate the performance of the MFI network and report the comparison results on the UCF101 and HMDB51 datasets. We utilize the pre-defined training/testing splits and protocols provided originally, and report the mean average accuracy over the three splits for HMDB51 and UCF101, respectively. As shown in Table II, only using 8 input frames, the proposed MFI network achieves 94.9% on UCF101 and 71.9% on HMDB51, respectively. Compared with the 2D CNNs based methods in the first compartment, our MFI network significantly outperforms the baseline method TSN by 8.5% (86.2% vs. 94.9%) on UCF101 and 7.2% (64.7% vs. 71.9%) on HMDB51. Our MFI network also gains the performance boost against other five 2D CNN based methods (i.e. Conv Fusion, Two-stream CNN, Two-stream TSN, StNet, and TSM) on both datasets. When utilizing 16 frames as input, our MFI network further obtains 95.6% on UCF101 and 73.3% on HMDB51 respectively. And compared with the 2D+3D or 3D CNNs based methods in the second compartment, MFI outperforms the most methods using the same or fewer number of input frames on both datasets, except for I3D. I3D-RGB uses 64 frames as input and two-stream I3D further utilizes optical flow information as the additional input modality. Therefore, the computing cost of I3D-RGB and two-stream I3D will be far more than us. Surprisingly, our MFI network with 16 input frames even slightly better than I3D-RGB on UCF101.

TABLE II
THE PERFORMANCE OF OUR MFI NETWORK ON ALL THREE SPLITS OF UCF101 AND HMDB51 COMPARED WITH STATE-OF-THE-ART METHODS.

Method	#Frames	UCF101	HMDB51
Two-stream CNN [2]	16+16	88.0	59.4
Two-stream TSN [13]	8+8	94.2	69.6
StNet [39]	7	93.5	-
TSM [25]	8	94.5	70.7
ECO [24]	92	93.6	68.0
STC-ReNeXt101 [40]	16	93.7	70.5
ARTNet [41]	16	94.3	70.9
I3D-RGB [7]	64	95.4	74.8
Two-steam I3D [7]	64+64	98.0	80.7
MFI(Ours)	8	94.9	71.9
MFI(Ours)	16	95.6	73.3

Note that, our MFI network has not utilized the optical flow information as the additional input modality, and the number of input frames is only 8 or 16. But our MFI network still gains comparable performance to the state-of-the-art methods. These results show the effectiveness of our MFI network to capture short-range motion features and learn long-range dependencies by feature interchange operation on UCF101 and HMDB51. This requires efficient methods like ours instead of additionally extracting the expensive optical-flow information, which is very computationally demanding, and limits the applications in the real-world.

C. Ablation study

In this section, we first testify the effectiveness of different components in our MFI network. Then we analyze the efficiency and performance of the proposed MFI network. All experiments in this section are conducted on Something-Something V1.

1) Components Effectiveness: To evaluate the effectiveness of each component in the proposed MFI network (i.e., CTI module and GRI module), we use 8 frames as input for training and compare the results of the baseline, the single module and the combination of both modules. As shown in Table

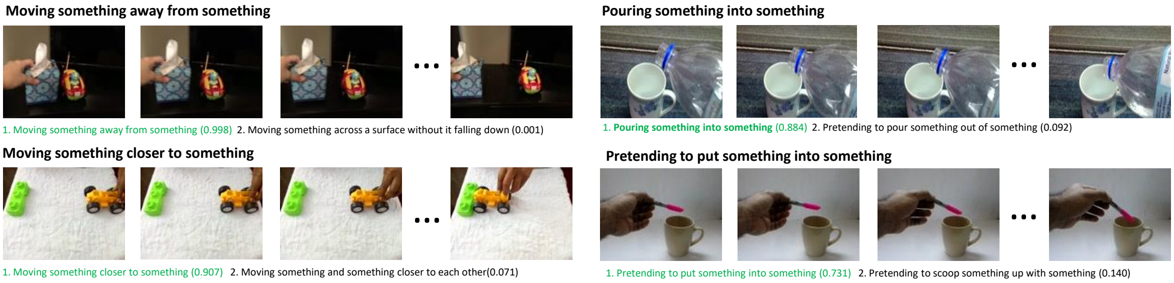
Moving something away from something
1. Moving something away from something (0.998) 2. Moving something across a surface without it falling down (0.001)
Pouring something into something
1. Pouring something into something (0.884) 2. Pretending to pour something out of something (0.092)
Moving something closer to something
1. Moving something closer to something (0.907) 2. Moving something and something closer to each other(0.071)
Pretending to put something into something
1. Pretending to put something into something (0.731) 2. Pretending to scoop something up with something (0.140)

Fig. 4. Some prediction examples on Something-Something V1. The top 2 predictions with green text indicating a correct prediction.

TABLE III
COMPONENTS EFFECTIVENESS OF MFI NETWORK.

Method	Val-Top1 (%)	Val-Top5 (%)
baseline(TSN)	19.7	46.6
GRI	38.2	67.2
CTI	42.8	71.3
MFI	43.9	73.9

III, compared to the baseline, GRI module learns long-range dependencies by performing graph-based feature interchange and gains 18.5% top1 accuracy improvement. Meanwhile, CTI module captures temporal difference related to motion information by utilizing channel-wise feature interchange and obtains 23.1% top1 accuracy improvement. Furthermore, better results can be produced by combining CTI module and GRI module together.

2) Efficiency Analysis: To analyze the efficiency of our MFI network, We compare the accuracy and model complexity of our MFI network with some state-of-the-art methods. As illustrated in Table IV, compared with the baseline model TSN, our MFI network achieves 2× higher accuracy while providing similar model complexity (0.02% higher FLOPs and 0.01% more parameters). And compared with ECO and I3D, our MFI network gains significant accuracy improvements with nearly 2× and 10× less FLOPs (33.6G vs 64G, 306G) and fewer parameters. Moreover, with similar model complexity, the proposed MFI network obtains slightly higher accuracy against TSM.

3) Performance Analysis: Finally, to analyze the performance of the proposed MFI network, we show some prediction examples of MFI network on Something-Something V1. As shown in Fig 4, the first column demonstrates that our MFI is able to correctly identifying actions that are closely related to the temporal order of frames. When reversing the temporal order of frames, the category “Moving something away from something” will be transformed into “Moving something closer to something”. And the second column shows our MFI network is also capable of correctly recognizing the “pretending” action category (e.g. Pretending to put something into something), where the actions contained in the short temporal range normally convey essential semantic information about the entire video action class.

TABLE IV
ACCURACY AND MODEL COMPLEXITY OF MFI NETWORK AND OTHER METHODS.

Model	#Frames	FLOPs	Param.	Acc.(%)
TSN [13]	8	33G	24.3M	19.7
	16	66G	24.3M	19.9
ECO [24]	16	64G	47.5M	41.4
I3D [7]	32	306G	28.0M	41.6
TSM [25]	8	33G	24.3M	43.4
	16	36G	24.3M	44.8
MFI	8	33.6G	24.6M	43.9
	16	67.2G	24.6M	45.5

V. CONCLUSION

In this paper, we propose a multi-range feature interchange (MFI) network for video action recognition, where the proposed channel-wise temporal interchange (CTI) module and graph-based regional interchange (GRI) module are used for encoding short-range motion features and learning long-range dependencies respectively. Furthermore, the proposed CTI module is embedded into the original bottleneck block in ResNet-50 to form a short-range temporal interchange (STI) block, we replace the original bottleneck blocks with STI blocks. Without any 3D convolution and additional optical flow modality, our MFI network gains comparable performance to the state-of-the-art methods on three video action recognition datasets using very limited computing costs.

ACKNOWLEDGMENT

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, 61773316, U1801262, and 61871470.

REFERENCES

[1] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[2] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[3] D. Wang, Y. Yuan, and Q. Wang, “Cross-modal message passing for two-stream fusion,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 1268–1272.

[4] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[5] Y. Yuan, D. Wang, and Q. Wang, “Memory-augmented temporal dynamic learning for action recognition,” *arXiv preprint arXiv:1904.13080*, 2019.

[6] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.

[7] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[8] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[9] A. Diba, V. Sharma, L. V. Gool, and R. Stiefelhagen, “Dynamonet: Dynamic action and motion network,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6192–6201.

[10] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, “Beyond gaussian pyramid: Multi-skip feature stacking for action recognition,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 204–212.

[11] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 3551–3558.

[12] X. Li, M. Chen, F. Nie, and Q. Wang, “Locality adaptive discriminant analysis.” in *IJCAI*, 2017, pp. 2201–2207.

[13] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, “Temporal segment networks: Towards good practices for deep action recognition,” in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 20–36.

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[15] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, “Temporal relational reasoning in videos,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 803–818.

[16] W. Zhang, J. Cen, and H. Zheng, “Temporal inception architecture for action recognition with convolutional neural networks,” in *International Conference on Pattern Recognition*, 2018, pp. 3216–3221.

[17] N. Srivastava, E. Mansimov, and R. Salakhudinov, “Unsupervised learning of video representations using lstms,” in *International conference on machine learning*, 2015, pp. 843–852.

[18] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[20] M. Lee, S. Lee, S. Son, G. Park, and N. Kwak, “Motion feature network: Fixed motion filter for action recognition,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 387–403.

[21] Z. Qiu, T. Yao, and T. Mei, “Learning spatio-temporal representation with pseudo-3d residual networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[22] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, “Human action recognition using factorized spatio-temporal convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4597–4605.

[23] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, “Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 305–321.

[24] M. Zolfaghari, K. Singh, and T. Brox, “Eco: Efficient convolutional network for online video understanding,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 695–712.

[25] J. Lin, C. Gan, and S. Han, “Tsm: Temporal shift module for efficient video understanding,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7083–7093.

[26] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, “Graph-based global reasoning networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 433–442.

[27] X. Wang and A. Gupta, “Videos as space-time region graphs,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 399–417.

[28] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, “Graph convolutional networks for temporal action localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7094–7103.

[29] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, “G-tad: Sub-graph localization for temporal action detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 156–10 165.

[30] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, “Aˆ 2-nets: Double attention networks,” in *Advances in Neural Information Processing Systems*, 2018, pp. 352–361.

[31] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek, “Videolstm convolves, attends and flows for action recognition,” *Computer Vision and Image Understanding*, vol. 166, pp. 41–50, 2018.

[32] J. Wang, X. Peng, and Y. Qiao, “Cascade multi-head attention networks for action recognition,” *Computer Vision and Image Understanding*, p. 102898, 2020.

[33] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[34] A. Diba, M. Fayyaz, V. Sharma, A. Hossein Karami, M. Mahdi Arzani, R. Yousefzadeh, and L. Van Gool, “Temporal 3d convnets using temporal transition layer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1117–1121.

[35] Z. Qiu, T. Yao, C.-W. Ngo, X. Tian, and T. Mei, “Learning spatio-temporal representation with local and global diffusion,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 056–12 065.

[36] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, “The” something something” video database for learning and evaluating visual common sense.” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, p. 3.

[37] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: a large video database for human motion recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2556–2563.

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

[39] D. He, Z. Zhou, C. Gan, F. Li, X. Liu, Y. Li, L. Wang, and S. Wen, “Stnet: Local and global spatial-temporal modeling for action recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8401–8408.

[40] A. Diba, M. Fayyaz, V. Sharma, M. Mahdi Arzani, R. Yousefzadeh, J. Gall, and L. Van Gool, “Spatio-temporal channel correlation networks for action classification,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 284–299.

[41] L. Wang, W. Li, W. Li, and L. Van Gool, “Appearance-and-relation networks for video classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1430–1439.