

# TEINet: Towards an Efficient Architecture for Video Recognition

Zhaoyang Liu,<sup>1\*</sup> Donghao Luo,<sup>2\*</sup> Yabiao Wang,<sup>2</sup> Limin Wang,<sup>1†</sup>  
Ying Tai,<sup>2</sup> Chengjie Wang,<sup>2</sup> Jilin Li,<sup>2</sup> Feiyue Huang,<sup>2</sup> Tong Lu<sup>1</sup>

<sup>1</sup>State Key Lab for Novel Software Technology, Nanjing University, China

<sup>2</sup>Youtu Lab, Tencent

## Abstract

Efficiency is an important issue in designing video architectures for action recognition. 3D CNNs have witnessed remarkable progress in action recognition from videos. However, compared with their 2D counterparts, 3D convolutions often introduce a large amount of parameters and cause high computational cost. To relieve this problem, we propose an efficient temporal module, termed as *Temporal Enhancement-and-Interaction (TEI Module)*, which could be plugged into the existing 2D CNNs (denoted by TEINet). The TEI module presents a different paradigm to learn temporal features by decoupling the modeling of channel correlation and temporal interaction. First, it contains a *Motion Enhanced Module (MEM)* which is to enhance the motion-related features while suppress irrelevant information (e.g., background). Then, it introduces a *Temporal Interaction Module (TIM)* which supplements the temporal contextual information in a channel-wise manner. This two-stage modeling scheme is not only able to capture temporal structure flexibly and effectively, but also efficient for model inference. We conduct extensive experiments to verify the effectiveness of TEINet on several benchmarks (e.g., Something-Something V1&V2, Kinetics, UCF101 and HMDB51). Our proposed TEINet can achieve a good recognition accuracy on these datasets but still preserve a high efficiency.

## 1 Introduction

Video understanding is one of the most important problems in computer vision (Simonyan and Zisserman 2014; Tran et al. 2015; Wang et al. 2016). Action recognition is a fundamental task in video understanding, as it is able to not only extract semantic information from videos, but also yield general video representations for other tasks such as action detection and localization (Feichtenhofer et al. 2018; Zhao et al. 2017). Unlike static images, the core problem of action recognition is how to model temporal information effectively. Temporal dimension typically exhibits a different property with respect to spatial domain. Modeling temporal

\*indicates equal contribution.

†Corresponding author (lmwang@nju.edu.cn).

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

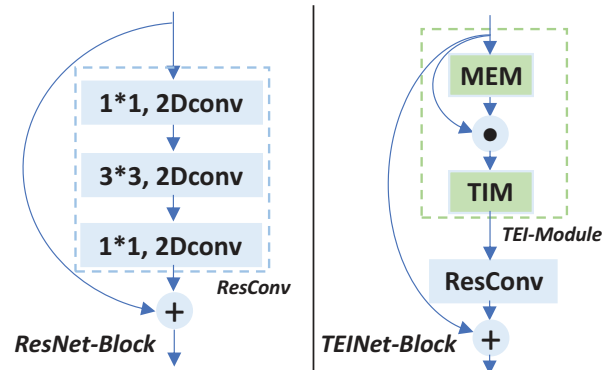


Figure 1: **TEINet building block.** We present an effective TEI module to decouple temporal modeling into a MEM to enhance motion-related features and a TIM capture temporal contextual information. This TEI module could be inserted into the 2D ResNet block to construct an efficient video architecture of TEINet.

information in a proper way is crucial for action recognition, which has aroused great interest of research.

Recently the convolutional networks (LeCun et al. 1998) have become the mainstream method in action recognition (Simonyan and Zisserman 2014; Carreira and Zisserman 2017; Tran et al. 2018). TSN (Wang et al. 2016) is an efficient method which ignores the constraint on temporal order information and only aggregates temporal information at the final classifier layer. To capture temporal information slowly and earlier, some new efficient 2D CNN based architectures are developed such as StNet (He et al. 2019) and TSM (Lin, Gan, and Han 2018). However, they involve some hand-crafted designs, which lacks a clear explanation and may be suboptimal for temporal modeling. 3D CNNs (Tran et al. 2015; Carreira and Zisserman 2017) are more principled architectures for directly learning spatiotemporal features from RGB frames. Unfortunately, this simple extension from a 2D convolution to its 3D version leads to a critical issue: it causes high computational cost when densely replacing 2D convolutions with 3D counterparts. Therefore we hope to devise a flexible temporal modeling module which

shares the capacity of learning spatiotemporal representations yet still keeps the efficiency of 2D CNNs.

Intuitively, temporal structure in video can benefit action recognition from multiple aspects. Firstly, motion information is able to help us focus on moving objects or people that are discriminative for action recognition. These discriminative features could be automatically determined for each input video. Secondly, the temporal evolution of visual features enables us to capture dynamic variation in videos and relate adjacent frame-level features for action recognition. Based on these analyses, we propose a new temporal modeling paradigm, termed as Enhance-and-Interact. This new design decouples the temporal module into two stages: first enhance discriminative features and then capture their temporal interaction. This unique design enables us to separately capture the channel-level correlation and temporal relation in a more principled and efficient way. It turns out that this separate modeling scheme is able to not only capture temporal structure flexibly and effectively, but also keeps a high inference efficiency in practice.

Specifically, we first present the *Motion Enhanced Module* (MEM), which utilizes motion information as a guide to focus on important features. To make this enhancement more efficient and effective, we squeeze the feature maps to only focus on channel-level importance and exploit temporal difference as an approximate motion map. Then, to capture the temporal interaction of adjacent frames, we present the *Temporal Interaction Module* (TIM), which model the local temporal variations of visual features. To control the model complexity and ensure the inference efficiency, we employ a temporal channel-wise convolution in a local time window. These two modules are plugged sequentially to yield a novel temporal module, namely *Temporal Enhancement-and-Interaction* (TEI module), which is **a generic building block and could be plugged into the existing 2D CNNs such as ResNets**, as illustrated in Figure 1.

In experiments, we verify the effectiveness of TEI block with the 2D ResNet on the large-scale datasets such as Kinetics (Kay et al. 2017) and Something-Something (Goyal et al. 2017). The final video architecture, coined as TEINet, obtains an evident performance improvement over previous approaches while is still able to keep fast inference speed. In particular, our TEINet achieves the state-of-the-art performance on the dataset of Something-Something, and comparable performance to previous 3D CNN based methods at a lower computational cost on the dataset of Kinetics. We also demonstrate the generalization ability of TEINet by fine-tuning on the datasets of UCF101 and HMDB51, where competitive recognition accuracy is also obtained. The main contribution in this work is summarized as follows:

- We present a new temporal modeling module, termed as TEI module, by decoupling the task of temporal feature learning into channel-level enhancement and local temporal interaction.
- The proposed TEINet is verified on various large-scale datasets, demonstrating that it is able to obtain an evident improvement over previous temporal modeling methods with a lower computational cost.

## 2 Related Work

**2D CNNs in Action Recognition.** Conventional 2D CNNs were extensively applied on action recognition in videos (Simonyan and Zisserman 2014; Feichtenhofer, Pinz, and Wildes 2016; Wang et al. 2016; Lin, Gan, and Han 2018; Gan et al. 2015). Two stream methods (Simonyan and Zisserman 2014; Feichtenhofer, Pinz, and Zisserman 2016; Zhang et al. 2016) regarded optical flow or motion vector as motion information to make up a temporal stream CNN. TSN (Wang et al. 2016) utilized average pooling to aggregate temporal information from set of sparsely-sampled frames. To improve the temporal reasoning ability of TSN, the TRN (Zhou et al. 2018) was proposed by focusing on the multi-scale temporal relations among sampled frames. To model temporal structure efficiently, TSM (Lin, Gan, and Han 2018) proposed a temporal shift module on the original feature map. Sharing the same motivation with TSM, our TEINet is also based on 2D backbones with high efficiency, but better at capturing temporal clues for video recognition.

**3D CNNs in Action Recognition.** 3D convolution (Tran et al. 2015; Carreira and Zisserman 2017) was a straightforward extension over 2D versions to learn the spatiotemporal representation directly from RGB. I3D (Carreira and Zisserman 2017) inflated all 2D convolution kernels into 3D convolution kernels and directly utilized the pre-trained weights on ImageNet (Deng et al. 2009). ARTNet (Wang et al. 2018a) improved the original 3D convolutions with higher-order relation modeling to explicitly capture motion. 3D convolution is natural and simple way for modeling temporal features, yet in practice with heavy computation. Unlike 3D CNNs, our TEINet resorts to a new temporal module purely based on 2D CNNs for video recognition.

**Efficient Temporal Modules.** Some efficient temporal models were proposed by using combination of 2D and 3D convolutions. ECO (Zolfaghari, Singh, and Brox 2018) combined the 2D convolution and 3D convolution into one network to achieve a balance between 2D CNNs and 3D CNNs. To decompose the optimization of spatial and temporal features, pseudo-3D convolution, e.g., P3D (Qiu, Yao, and Mei 2017), S3D (Xie et al. 2018) and R(2+1)D (Tran et al. 2018), decomposed the spatio-temporal 3D convolution into a spatial 2D convolution and a temporal 1D convolution. Our TEINet integrates a new temporal block into a purely 2D backbone to endow network with the ability to model temporal structure in videos.

**Attention in Action Recognition.** The attention mechanism (Hu, Shen, and Sun 2018; Li, Hu, and Yang 2019) were widely used in image classification, which can boost performance using a small portion of extra parameters. Similarly, there are some works (Wang et al. 2018b; Girdhar and Ramanan 2017) related to attention in action recognition. Non-local network formulates the non-local mean operation as Non-local block to capture the long-range dependencies in video. Motion enhanced module (MEM) in our method differs from these attention methods. MEM constructs the temporal attention weights by local motion tendency which can be trained by end-to-end without using extra supervision and give a sizable boost in accuracy.

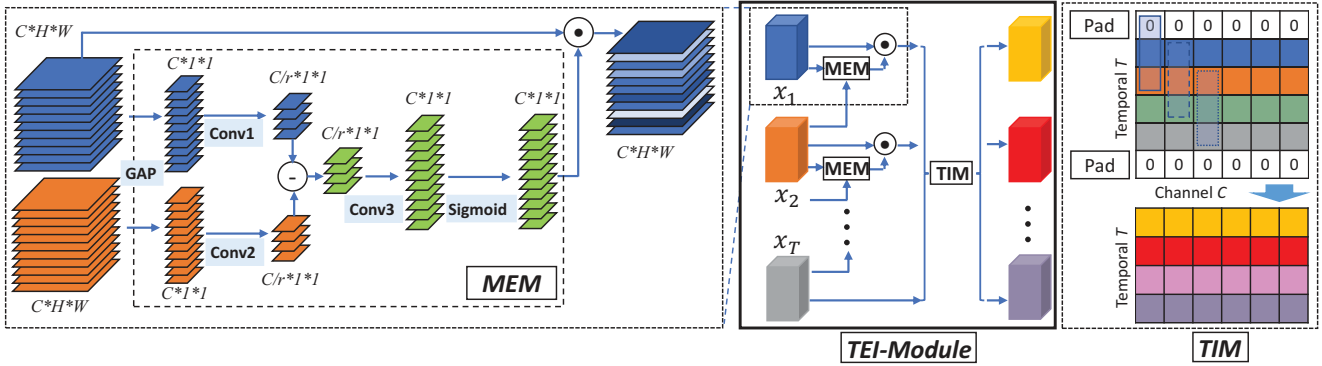


Figure 2: The pipeline of TEI module. We show motion enhanced module (MEM) in the left and temporal interaction module (TIM) in the right. The  $\odot$  denotes element-wise multiplication, and  $\ominus$  denotes element-wise subtraction. Notably, in TIM, we use different box to represent kernel weights, which means each channel do not share kernel weights.

### 3 Method

We will introduce our proposed TEI module in this section. First, we describe motion enhanced module and explain how to learn a channel level attention weights. Then we present the technical details of temporal interaction module. Finally we combine these two modules as a TEINet building block and integrate this block into the off-the-shelf architecture of 2D CNN.

#### 3.1 Motion Enhanced Module

In action recognition, spatial features can only provide partial information for action recognition. It is well established that motion information is an crucial cue for understanding human behavior in videos. Consequently, we first design a *Motion Enhanced Module* (MEM) to focus on motion-salient features while suppress the irrelevant information at background.

Our method is to enhance the motion-related features in a channel-wise way by using the temporal difference of adjacent frame level features. To decrease the computational cost, we first construct a global representation for each channel and then perform feature enhancement in a channel level. As depicted in Figure 2, given an input sequence  $X = \{x_1, x_2, \dots, x_T\}$  in which  $x_t \in \mathbb{R}^{C \times H \times W}$ , We first aggregate feature map  $x_t$  across their spatial dimensions ( $H \times W$ ) by using global average pooling, which produces  $\hat{x}_t \in \mathbb{R}^{C \times 1 \times 1}$ . Then, these pooled features go through subsequent processing operations to generate the channel importance weights.

Basically, we observe that the overall appearance information varies gradually and slowly over time. The pixel values in motion salient regions would change more quickly than those in static regions. In practice, we exploit the feature difference between adjacent frames to approximately represent the motion saliency. To reduce the model complexity, the  $\hat{x}_t$  and  $\hat{x}_{t+1}$  are fed into two different 2D convolutions whose kernel size of  $1 \times 1$  in which the channels of  $\hat{x}_t$  will be compressed. This dimension reduction and difference calculation can be formulated as:

$$s_t = \text{Conv1}(\hat{x}_t, W_\theta) - \text{Conv2}(\hat{x}_{t+1}, W_\phi). \quad (1)$$

Here  $W_\theta$  and  $W_\phi$  are learnable parameters of the convolutions that reduce the number of channels in  $\hat{X}$  from  $C$  to  $\frac{C}{r}$ . In our experiments, the reduction ratio  $r$  is set to 8.

Then another 2D convolution is applied on  $s_t$ , which aims to recover the channel dimension of  $s_t$  as same as input  $s_t$ . The attention weights are obtained by:

$$\hat{s}_t = \sigma(\text{Conv3}(s_t, W_\varphi)), \quad (2)$$

where  $\sigma(*)$  denotes a sigmoid function and  $W_\varphi$  are learnable parameters of Conv3. Finally we obtain attention weights  $\hat{s} \in \mathbb{R}^{C \times 1 \times 1}$  for different channels. We utilize channel-wise multiplication to enhance motion-salient features:

$$u_t = \hat{s}_t \cdot x_t, \quad (3)$$

where  $t \in [1, T - 1]$  and  $u_t$  is our final enhanced feature map. To keep the temporal scale consistent with input  $X$ , we simply copy the  $x_T$  as  $u_T$ , namely,  $u_T = x_T$ .

**Discussion.** We have noticed that our MEM is similar to SE module in (Hu, Shen, and Sun 2018). However, the essential difference between SE module and MEM is that SE module is a kind of self attention mechanism by using its own global feature to calibrate the different channels, while our MEM is a motion-aware attention module by enhancing the motion-related features. To prove the effectiveness of MEM, we conduct the comparative experiments in section 4.3. Under the same setting, our MEM is better at enhancing temporal features for action recognition than SE module in video dataset.

#### 3.2 Temporal Interaction Module

In MEM, we enhance the motion-related features, but our model is still incapable of capturing temporal information in a local time window, namely the temporal evolution of visual pattern over time. Consequently, we propose the *Temporal Interaction Module* (TIM) which aims to capture temporal contextual information at a low computational cost. More specifically, we here use a channel-wise convolution to learn the temporal evolution for each channel independently, which preserves low computational complexity for model design.



As illustrated in Figure 2, given an input  $U = \{u_1, u_2, \dots, u_T\}$ , we first transform its shape from  $U^{T \times C \times H \times W}$  to  $\hat{U}^{C \times T \times H \times W}$  (denoted by  $\hat{U}$  to avoid ambiguity). Then we apply the channel-wise convolution to operate on  $\hat{U}$  as follows:

$$Y_{c,t,x,y} = \sum_i V_{c,i} \cdot \hat{U}_{c,t+i,x,y}, \quad (4)$$

where  $V$  is the channel-wise convolutional kernel and  $Y_{c,t,x,y}$  is the output after temporal convolution. The channel-wise convolution tremendously decreases the computation costs comparing with 3D convolution. In our setting, the kernel size of the channel-wise convolution is  $3 \times 1 \times 1$ , which implies the features are only interacting with features in adjacent time, but the temporal receptive fields will gradually grow when feature maps pass through deeper layers of network. After convolution, we will transform the shape of output  $Y$  back to  $T \times C \times H \times W$ . The parameters of vanilla 3D convolution is  $C_{out} \times C_{in} \times t \times d \times d$ , and the temporal 1D convolution in (Tran et al. 2018) is  $C_{out} \times C_{in} \times t$ , but the parameters of TIM is  $C_{out} \times 1 \times t$ . The number of parameters in TIM is greatly reduced when compared with other temporal convolutional operators.

**Discussion.** We figure out our TIM is related to the recent proposed TSM (Lin, Gan, and Han 2018). In fact, TSM could be viewed as a channel-wise temporal convolution, where temporal kernel is fixed as  $[0, 1, 0]$  for non shift,  $[1, 0, 0]$  for backward shift, and  $[0, 0, 1]$  for forward shift. Our TIM generalizes the TSM operation into a flexible module with a learnable convolutional kernel. In experiment, we find that this learnable scheme is more effective than random shift to capture temporal contextual information for action recognition.

### 3.3 TEINet

After introducing the MEM and TIM, we are ready to describe how to build the temporal enhancement-and-interaction block (TEI) and integrate it into the existing network architecture. As shown in Figure 1, the TEI module is composed of MEM and TIM introduced above, which could be implemented efficiently. First the input feature maps will be fed into MEM to learn attention weights for different channels, which aims to enhance the motion-related features. Then the enhanced features will be fed into the TIM to capture temporal contextual information. Our TEI module is a generic and efficient temporal modeling module, which could be plugged into any existing 2D CNN to capture temporal information, and the resulted network is called *Temporal Enhancement-and-Interaction Network* (TEINet).

Our TEI module is directly inserted into the 2D CNN backbone, while other methods (Tran et al. 2018; Qiu, Yao, and Mei 2017; Xie et al. 2018) replace the 2D convolutions with more expensive 3D convolutions or (2+1)D convolutions. This new integration method is able to not only use the pre-trained ImageNet model for initialization but also bring a smaller number of extra computational FLOPs compared with 3D CNNs. In our experiments, to trade off between performance and computational cost, we instantiate the temporal enhancement-and-interaction network (TEINet) using

ResNet-50 (He et al. 2016) as backbone. We conduct extensive experiments to figure out the optimal setting of TEINet for action recognition in Section. 4.

**Discussion.** Our paper proposed enhancement-and-interaction is a factorized modeling method to endow network with a strong ability to learn the temporal features in videos. We find that our module is effective for both types of video datasets: motion dominated one such as Something-Something V1&V2 and appearance dominated one such as Kinetics-400. MEM and TIM focus on different aspects when capturing temporal information, where MEM aims to learn channel level importance weights and TIM tries to learn temporal variation pattern of adjacent features. These two modules are cooperative and complementary to each other as demonstrated in Table 1a.

## 4 Experiments

### 4.1 Datasets

**Something-Something V1&V2.** (Goyal et al. 2017) is a large collection of video clips containing daily actions interacting with common objects. It tries to focus on motion itself without differentiating manipulated objects. V1 includes 108499 video clips, and V2 includes 220847 video clips. They both have 174 classes.

**Kinetics-400.** (Kay et al. 2017) is a large-scale dataset in action recognition, which contains 400 human action classes, with at least 400 video clips for each class. Each clip is collected from YouTube videos and then trimmed to around 10s. The newest version of Kinetics has updated to Kinetics-700 which approximately includes 650k video clips that covers 700 human action classes. For fair comparison with previous methods, we conduct experiments on Kinetics-400.

**UCF101 and HMDB51.** Finally, to verify the generalization ability to transfer to smaller scale datasets, we report the results on the datasets of UCF101 (Soomro, Zamir, and Shah 2012) and HMDB51 (Kuehne et al. 2011). The UCF101 contains 101 categories with around 13k videos, while HMDB51 has about 7k videos spanning over 51 categories. On UCF101 and HMDB51, we follow the common practice that reports the accuracy by averaging over three splits. Different from Something-Something, the datasets of Kinetics-400, UCF101 and HMDB51 are less sensitive to temporal relationship.

### 4.2 Experimental Setup

We here choose the ResNet-50 as our backbone for the trade off between performance and efficiency. Unless specified, our model is pre-trained on ImageNet (Deng et al. 2009).

**Training.** We applied a similar pre-processing method to (Wang et al. 2018b): first resizing the shorter side of raw images to 256 and then employing a center cropping and scale-jittering. Before being fed into the network, the images will be resized to  $224 \times 224$ . In our model, We attempt to stack 8 frames or 16 frames as a clip. On the Kinetics dataset, we train our models for 100 epochs in total, starting with a learning rate of 0.01 and reducing to its  $\frac{1}{10}$  at 50, 75, 90 epochs. For fair comparisons with the state-of-the-art models, we follow the testing strategy in (Lin, Gan, and

model	Top-1	Top-5
Res50+TSN	19.7%	46.6%
Res50+TSM	43.4%	73.2%
Res50+MEM	33.5%	61.5%
Res50+TIM	46.1%	74.7%
Res50+SE+TIM	46.1%	75.2%
Res50+MEM+TIM	<b>47.4%</b>	<b>76.6%</b>

(a) Exploration on MEM and TIM, and comparison with other baseline methods.

stage	Top-1	Top-5
res <sub>2</sub>	41.6%	70.1%
res <sub>3</sub>	43.1%	72.1%
res <sub>4</sub>	<b>45.4%</b>	<b>74.6%</b>
res <sub>5</sub>	45.3%	74.3%

(b) The TEI blocks in different stage of ResNet-50

stages	Blocks	Top-1	Top-5
res <sub>5</sub>	3	45.3%	74.3%
res <sub>4-5</sub>	9	46.7%	76.3%
res <sub>3-5</sub>	13	47.3%	75.2%
res <sub>2-5</sub>	16	<b>47.4%</b>	<b>75.8%</b>

(c) The number of TEI block inserted into ResNet-50.

Table 1: Ablation studies on Something-Something V1.

Method	Frame	Params	FLOPs	Latency	Throughput	Sthv1
I3D (Carreira et al. 2017)	64	35.3M	360G	165.3ms	6.1 vid/s	41.6%
ECO <sub>16f</sub> (Zolfaghari et al. 2018)	16	47.5M	64G	30.6ms	45.6 vid/s	41.4%
TSN (Wang et al. 2016)	8	24.3M	33G	15.5ms	81.5 vid/s	19.7%
TSM (Lin et al. 2018)	8	24.3M	33G	17.4ms	77.4 vid/s	43.4%
TSM (Lin et al. 2018)	16	24.3M	65G	29.0ms	39.5 vid/s	44.8%
Res50+TIM	8	24.3M	33G	20.1ms	61.6 vid/s	46.1%
TEINet	8	30.4M	33G	36.5ms	46.9 vid/s	47.4%
Res50+TIM	16	24.3M	66G	34.9ms	31.4 vid/s	48.5%
TEINet	16	30.4M	66G	49.5ms	24.2 vis/s	49.9%

Table 2: Quantitatively analysis on latency and throughput Something-Something V1. "vid/s" represents videos per second. The larger latency and the smaller throughput represent higher efficiency.

Han 2018), which uniformly samples 8 or 16 frames from the consecutive 64 frames randomly sampled in each video. We observe that the duration of most videos in Something-Something V1&V2 normally has less than 64 frames. Thus we use the similar strategy to TSN (Wang et al. 2016) to train our model. Specifically, we uniformly sample the 8 or 16 frames from each video. On Something-Something V1&V2, We train the TEINet for 50 epochs starting with a learning rate 0.01 and reducing it by a factor of 10 at 30, 40, 45 epochs. For all of our experiments, we utilize SGD with momentum 0.9 and weight decay of 1e-4 to train our TEINet on Tesla M40 GPUs using a mini batch size of 64.

**Inference.** We follow the widely used settings in (Wang et al. 2018b; Lin, Gan, and Han 2018): resizing shorter side to 256 and taking 3 crops (left, middle, right) in each frame. Then we uniformly sample 10 clips in each video and compute the classification scores for all clips individually. The final prediction will be obtained by utilizing the average pooling to aggregate the scores of 10 clips.

### 4.3 Ablation Studies

This section provides ablation studies on TEI module design and integration with ResNet50 on the Something-Something V1 dataset. In this section, we report the experimental results using the testing scheme of center crop and one clip, the results are summarized in Table 1.

**Study on MEM and TIM.** We first conduct a separate study on the effect of each individual module (MEM or TIM) on action recognition. We find that the TIM is able to yield a better recognition accuracy than MEM (46.1% vs. 33.5%), indicating that temporal contextual information is more important for action recognition in the Something-Something

dataset. Then, we compare our TIM with other efficient temporal modeling baselines such as TSN and TSM, which demonstrates that our TIM is more effective than these baseline methods. Finally, we compare the performance of MEM with SE on action recognition, and we see that MEM+TIM is better than SE+TIM by 1.3%, which confirms our motivation that motion-aware attention is better at capturing discriminative temporal features for action recognition.

**Which stage to insert TEI blocks.** As shown in Table 1b, we find that a clear performance improvement will be obtained when inserting TEI block in the later stages. It is worth noting that res<sub>4</sub> has 3 more blocks than res<sub>5</sub>, but integration at both locations achieves a similar result. The temporal modeling based on higher level features may be more beneficial to recognition, which agrees with the findings from (Xie et al. 2018).

**The number of TEI block inserted into network.** Efficiency is an important issue and sometimes we may focus on improving recognition accuracy with a limited extra computation consumption. We here expect to figure out how many TEI blocks can obtain a trade-off performance. Specifically, We attempt to gradually add TEI blocks from res<sub>5</sub> to res<sub>2</sub> in ResNet50. As shown in Table 1c where we use the same inference settings as Table 1a, we can boost the performance by inserting more TEI blocks. We also see that res<sub>2-5</sub> only outperforms res<sub>4-5</sub> by 0.7%, but with extra 7 TEI blocks. Therefore, in practice, we recommend to use TEI block simply in stages of res<sub>4-5</sub> as it is more efficient. But our default choice of the remaining experiments are ready to use the ResNet50 equipped with TEI blocks in all stages.

**Analysis on runtime.** The runtime of model has also drawn considerable attention from researchers in recent years. Sev-

Method	Backbone	Pre-train	Frames	FLOPs	Val Top-1	Test Top-1
TSN-RGB (Wang et al. 2016)	ResNet2D-50	ImgNet	$8f$	33G	19.7%	-
TRN-Multiscale-RGB (Zhou et al. 2018)	BNInception		$8f$	33G	34.4%	33.6%
TRN-Multiscale-RGB (Zhou et al. 2018)	ResNet2D-50	ImgNet	$8f$	33G	38.9%	-
TRN-Multiscale-2Stream (Zhou et al. 2018)*	BNInception		$8f + 8f$	-	42.0%	40.7%
S3D-G-RGB (Xie et al. 2018)	Inception	ImgNet	$64f$	71G	48.2%	-
I3D-RGB (Wang and Gupta 2018)	ResNet3D-50			306G	41.6%	-
NL I3D-RGB (Wang and Gupta 2018)	ResNet3D-50	ImgNet+K400	$32f \times 2$	334G	44.4%	-
NL I3D+GCN-RGB (Wang and Gupta 2018)	ResNet3D-50+GCN			606G	46.1%	45.0%
ECO-RGB (Zolfaghari et al. 2018)			$16f$	64G	41.6%	-
ECO-RGB (Zolfaghari et al. 2018)	BNIncep+Res3D-18	K400	$92f$	267G	46.4%	-
ECO <sub>E<sub>n</sub></sub> Lite-2Stream (Zolfaghari et al. 2018)*			$92f + 92f$	-	49.5%	43.9
TSM-RGB (Lin, Gan, and Han 2018)			$8f$	33G	43.4%	-
TSM-RGB (Lin, Gan, and Han 2018)			$16f$	65G	44.8%	-
TSM <sub>E<sub>n</sub></sub> -RGB (Lin, Gan, and Han 2018)	ResNet2D-50	ImgNet+K400	$16f + 8f$	98G	46.8%	-
TSM-2Stream (Lin, Gan, and Han 2018)*			$16f + 16f$	-	50.2%	47.0
			$8f$	33G	<b>47.4%</b>	-
TEINet-RGB	ResNet2D-50	ImgNet	$8f \times 10$	990G	48.8%	-
			$16f$	66G	<b>49.9%</b>	-
			$16f \times 10$	1980G	51.0%	44.7%
TEINet <sub>E<sub>n</sub></sub> -RGB	ResNet2D-50	ImgNet	$16f + 8f$	99G	<b>52.5%</b>	<b>46.1%</b>

Table 3: Comparison with the state-of-the-art on Something-Something V1. The \* represents that they use two stream method and can not directly compare with our method.

Method	Val	Test
TSN <sub>16f×10</sub> (Wang et al. 2016)	30.0%	-
TRN-RGB <sub>8f</sub> (Zhou et al. 2018)	48.8%	50.9%
TRN-2Stream <sub>8f</sub> (Zhou et al. 2018)	55.5%	83.1%
TSM-RGB <sub>8f×10</sub> (Lin et al. 2018)	59.1%	-
TSM-RGB <sub>16f×10</sub> (Lin et al. 2018)	59.4%	60.4%
TSM-2Stream <sub>16f×10</sub> (Lin et al. 2018)	64.0%	64.3%
TEINet-RGB <sub>8f</sub>	61.3%	-
TEINet-RGB <sub>8f×10</sub>	64.0%	62.7%
TEINet-RGB <sub>16f</sub>	62.1%	-
TEINet RGB <sub>16f×10</sub>	64.7%	63.0%
TEINet RGB <sub>16f+8f</sub>	<b>66.5%</b>	<b>64.6%</b>

Table 4: Comparison with the state-of-the-art on Something-Something V2. The subscript  $8f \times 10$  denotes we sample 10 clips and each clip contain 8 frames.

eral experiments are conducted on Something-Something V1 to manifest the latency and throughput for our models. For the fair comparisons with other models, we follow the inference settings in (Lin, Gan, and Han 2018) by using a single NVIDIA Tesla P100 GPU to measure the latency and throughput. We use a batch size of 1 to measure the latency and a batch size of 16 to measure the throughput. Data loading time is not considered in this experiment. As shown in Table 2, Our models achieve the acceptable latency and Throughput comparing with other models.

#### 4.4 Comparison with the State of the Art

**Results on Something-Something V1.** We compare our TEINet with the current state-of-the-art models in Table 3. It is worth noting that our proposed models are only pre-

trained on ImageNet. For fair and detailed comparison, the results of TEINet apply center crop when sampling 1 clip, and 3 crops when sampling 10 clips. We notice that our TEINets dramatically outperform TSN (Wang et al. 2016), which demonstrates the effectiveness of TEI Block. When using 16 frames as input our proposed TEINet outperforms TSM (Lin, Gan, and Han 2018) by 5.1% on validation set and even achieves superior performance to TSM<sub>E<sub>n</sub></sub> which ensembles the results of 8 frames and 16 frames. As manifested in Table 3, Our TEINet<sub>E<sub>n</sub></sub> which has the same setting as TSM<sub>E<sub>n</sub></sub> can surpass all existing RGB or RGB+Flow based models on Something-Something V1. When it comes to computational costs, we also list FLOPs for most models, We find that our model achieves the superior performance with reasonable FLOPs during testing. More analyses on runtime comparing with other models have been mentioned in Section 4.3.

**Results on Something-Something V2.** As shown in Table 4, We report the results on Something-Something V2 which is a new release of V1. The training setting and inference protocol of Table 4 are consistent with Table 3. Our proposed TEINets obtain the similar performance gain on Something-Something V2 by only using RGB as input. The TEINet<sub>8f</sub> even achieves 61.3% and outperforms TSM<sub>16f×10</sub> as inputs by 1.9%. Furthermore, our proposed TEINet<sub>16f+8f</sub> which ensembles the models using 16 frames and 8 frames as inputs outperforms TSM-2Stream by 2.5%, and achieves superior performance to the previous state-of-the-art models, which demonstrates that our TEINet is able to capture temporal features on this motion sensitive dataset.

**Results on Kinetics-400.** The Kinetics is currently the most popular dataset in action recognition, due to its large num-

Method	Backbone	Pre-train	GFLOPs×views	Top-1	Top-5
I3D <sub>64f</sub> (Carreira et al. 2017)	Inception V1	ImgNet	108×N/A	72.1%	90.3%
I3D <sub>64f</sub> +TSN (Wang et al. 2019)	Inception V1	ImgNet	108×N/A	73.5%	91.6%
ARTNet <sub>16f</sub> +TSN (Wang et al. 2018a)	ResNet-18	From Scratch	23.5×250	70.7%	89.3%
NL+I3D <sub>32f</sub> (Wang et al. 2018b)	ResNet-50	ImgNet	70.5×30	74.9%	91.6%
NL+I3D <sub>128f</sub> (Wang et al. 2018b)	ResNet-101	ImgNet	359×30	77.7%	93.3%
Slowfast (Feichtenhofer et al. 2018)	ResNet-50	From Scratch	36.1×30	75.6%	92.1%
Slowfast (Feichtenhofer et al. 2018)	ResNet-101	From Scratch	106×30	77.9%	93.2%
NL+Slowfast (Feichtenhofer et al. 2018)	ResNet-101	From Scratch	234×30	<b>79.8%</b>	<b>93.9%</b>
LGD-3D <sub>128f</sub> (Qiu et al. 2019)	ResNet-101	ImgNet	N/A×N/A	79.4%	94.4%
TSN (Wang et al. 2016)	Inception V3	ImgNet	3.2×250	72.5%	90.2%
ECO <sub>En</sub> (Zolfaghari, Singh, and Brox 2018)	BNIncep+Res3D-18	From Scratch	N/A×N/A	70.7%	89.4%
R(2+1)D <sub>32f</sub> (Tran et al. 2018)	ResNet-34	Sports-1M	152×10	74.3%	91.4%
S3D-G <sub>64f</sub> (Xie et al. 2018)	Inception V1	ImgNet	71.4×30	74.7%	93.4%
StNet <sub>25f</sub> (He et al. 2019)	ResNet-50	ImgNet	189.3×1	69.9%	-
TSM <sub>16f</sub> (Lin, Gan, and Han 2018)	ResNet-50	ImgNet	65×30	74.7%	91.4%
TEINet <sub>8f</sub>	ResNet-50	ImgNet	33×30	74.9%	91.8%
TEINet <sub>16f</sub>	ResNet-50	ImgNet	66×30	76.2%	92.5%

Table 5: Comparison with the state-of-the-art models on Kinetics-400. Similar to (Feichtenhofer et al. 2018), we report the inference cost by computing the GFLOPs (of a single view) × the number of views (temporal clips with spatial crops). The subscript 8f denotes each clip contains 8-frame and N/A denotes the numbers are not available for us.

Method	UCF	HMDB
TSN-RGB (Wang et al. 2016)	93.2%	-
I3D-RGB (Carreira et al. 2017)	95.6%	74.8%
P3D-RGB (Qiu, Yao, and Mei 2017)	88.6%	-
S3D-G-RGB (Xie et al. 2018)	96.8%	75.9%
R(2+1)D-RGB (Tran et al. 2018)	96.8%	74.5%
TSM-RGB (Lin et al. 2018)	96.0%	73.2%
ECO <sub>En</sub> (Zolfaghari et al. 2018)	94.8%	72.4%
ARTNet-RGB (Wang et al. 2018a)	94.3%	70.9%
StNet-RGB (He et al. 2019)	93.5%	-
TEINet-RGB	96.7%	72.1%

Table 6: Comparison with the state-of-the-art models on UCF101 and HMDB51. The results are followed common practice that reports accuracy by averaging over all 3 splits. For fair comparison, we only list the models using RGB as inputs.

bers of videos and various categories. The results are summarized in Table 5. The upper part of Table 5 lists the current state-of-the-art models based on 3D convolutions, which are with expensive computational costs. The middle part of Table 5 lists several slightly lightweight models which are mainly composed of 2D convolutions or a few 3D convolutions in network. Notably, our models based on 2D ResNet-50 only utilize ImageNet as pre-training dataset. We here only list the models only using RGB as inputs to perform comparisons. As shown in Table 5, our TEINet obtain a better performance gain among lightweight models. Meanwhile, our models even achieve competitive performance when comparing with computationally expensive models. For example, the TEINet using 16-frame as inputs outperforms NL I3D using 32-frame as inputs by 1.3%.

**Transferring to UCF101 and HMDB51.** To verify the generalization of TEINet on smaller datasets, we evaluate the performance for our models on UCF101 and HMDB51. We

fine-tune our TEINet on the UCF10 and HMDB51 datasets using model pre-trained on Kinetics-400, and report the performance using 10 clips and 3 crops per video. We here only list our model using 16-frame as inputs. As shown in Table 6, our proposed TEINet also achieves competitive performance when comparing with I3D-RGB and R(2+1)D-RGB, which demonstrates the the generalization ability of our method.

## 5 Conclusion

In this work, we have proposed an efficient temporal modeling method, i.e., TEINet, to capture temporal features in video frames for action recognition. The vanilla ResNet can be converted into TEINet by inserting the TEI blocks which are composed of a motion enhanced module (MEM) and a temporal interaction module (TIM). The MEM focuses on enhancing the motion-related features by calculating temporal attention weights, and TIM is to learn the temporal contextual features with a channel-wise temporal convolution. We conducted a series of empirical studies to demonstrate the effectiveness of TEINet for action recognition in videos. The experimental results show that our method has achieved the state-of-the-art performance on the Something-Something V1&V2 dataset and competitive performance on the Kinetics dataset with a high efficiency.

## 6 Acknowledgments

This work is supported by the National Science Foundation of China (No. 61921006), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

## References

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



- Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2009*, 248–255.
- Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2018. Slowfast networks for video recognition. *CoRR* abs/1812.03982.
- Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2016. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, 3468–3476.
- Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 1933–1941.
- Gan, C.; Wang, N.; Yang, Y.; Yeung, D.-Y.; and Hauptmann, A. G. 2015. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2568–2577.
- Girdhar, R., and Ramanan, D. 2017. Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 34–45.
- Goyal, R.; Kahou, S. E.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haanel, V.; Fründ, I.; Yianilos, P.; Mueller-Freitag, M.; Hoppe, F.; Thureau, C.; Bax, I.; and Memisevic, R. 2017. The “something something” video database for learning and evaluating visual common sense. In *IEEE International Conference on Computer Vision, ICCV 2017*, 5843–5851.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 770–778.
- He, D.; Zhou, Z.; Gan, C.; Li, F.; Liu, X.; Li, Y.; Wang, L.; and Wen, S. 2019. Stnet: Local and global spatial-temporal modeling for action recognition. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019*, 8401–8408.
- Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 7132–7141.
- Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The kinetics human action video dataset. *CoRR* abs/1705.06950.
- Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T. A.; and Serre, T. 2011. HMDB: A large video database for human motion recognition. In *IEEE International Conference on Computer Vision, ICCV 2011*, 2556–2563.
- LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.
- Li, X.; Hu, X.; and Yang, J. 2019. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *CoRR* abs/1905.09646.
- Lin, J.; Gan, C.; and Han, S. 2018. Temporal shift module for efficient video understanding. *CoRR* abs/1811.08383.
- Qiu, Z.; Yao, T.; Ngo, C.; Tian, X.; and Mei, T. 2019. Learning spatio-temporal representation with local and global diffusion. *CoRR* abs/1906.05571.
- Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *IEEE International Conference on Computer Vision, ICCV 2017*, 5534–5542.
- Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, 568–576.
- Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR* abs/1212.0402.
- Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015*, 4489–4497.
- Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 6450–6459.
- Wang, X., and Gupta, A. 2018. Videos as space-time region graphs. In *Computer Vision - ECCV 2018*, 413–431.
- Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Gool, L. V. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *Computer Vision - ECCV 2016*, 20–36.
- Wang, L.; Li, W.; Li, W.; and Gool, L. V. 2018a. Appearance-and-relation networks for video classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 1430–1439.
- Wang, X.; Girshick, R. B.; Gupta, A.; and He, K. 2018b. Non-local neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 7794–7803.
- Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Gool, L. V. 2019. Temporal segment networks for action recognition in videos. *IEEE Trans. Pattern Anal. Mach. Intell.* 41(11):2740–2755.
- Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Computer Vision - ECCV 2018*, 318–335.
- Zhang, B.; Wang, L.; Wang, Z.; Qiao, Y.; and Wang, H. 2016. Real-time action recognition with enhanced motion vector cnns. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 2718–2726.
- Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In *IEEE International Conference on Computer Vision, ICCV 2017*, 2933–2942.
- Zhou, B.; Andonian, A.; Oliva, A.; and Torralba, A. 2018. Temporal relational reasoning in videos. In *Computer Vision - ECCV 2018*, 831–846.
- Zolfaghari, M.; Singh, K.; and Brox, T. 2018. ECO: efficient convolutional network for online video understanding. In *Computer Vision - ECCV 2018*, 713–730.