Temporal Distinct Representation Learning for Action Recognition

Junwu Weng*[1,3], Donghao Luo*[2], Yabiao Wang[2], Ying Tai[2], Chengjie Wang[2], Jilin Li[2], Feiyue Huang[2], Xudong Jiang[3], and Junsong Yuan[4]

[1] Tencent AI Lab
[2] Tencent Youtu Lab
[3] School of EEE, Nanyang Technological University
[4] Department of CSE, The State University of New York, Buffalo
{calweng, michaelluo, caseywang, yingtai, jasoncjwang, jerolinli, garyhuang}@tencent.com
exdjiang@ntu.edu.sg jsyuan@buffalo.edu

Abstract. Motivated by the previous success of Two-Dimensional Convolutional Neural Network (2D CNN) on image recognition, researchers endeavor to leverage it to characterize videos. However, one limitation of applying 2D CNN to analyze videos is that different frames of a video share the same 2D CNN kernels, which may result in repeated and redundant information utilization, especially in the spatial semantics extraction process, hence neglecting the critical variations among frames. In this paper, we attempt to tackle this issue through two ways. 1) Design a sequential channel filtering mechanism, i.e., Progressive Enhancement Module (PEM), to excite the discriminative channels of features from different frames step by step, and thus avoid repeated information extraction. 2) Create a Temporal Diversity Loss (TD Loss) to force the kernels to concentrate on and capture the variations among frames rather than the image regions with similar appearance. Our method is evaluated on benchmark temporal reasoning datasets Something-Something V1 and V2, and it achieves visible improvements over the best competitor by 2.4% and 1.3%, respectively. Besides, performance improvements over the 2D-CNN-based state-of-the-arts on the large-scale dataset Kinetics are also witnessed.

Keywords: Video Representation Learning, Action Recognition, Progressive Enhancement Module, Temporal Diversity Loss

1 Introduction

Owing to the computer vision applications in many areas like intelligent surveillance and behavior analysis, how to characterize and understand videos becomes an intriguing topic in the computer vision community. To date, a large number of deep learning models [20, 23, 10, 33, 25, 29, 12, 13, 14] have been proposed to

* Equal contribution. This work is done when Junwu Weng is an intern at Youtu Lab.

analyze videos. The RNN-based models [29, 30] are common tools for sequence modeling for its sequential nature of visual representation processing, by which the order of a sequence can be realized. However, in these models the spatial appearance and temporal information are learned separately. Motivated by the success in image recognition, Convolutional Neural Network (CNN) becomes popular for video analysis. 3D CNNs [25, 26, 4, 20] are widely used in video analysis as they can jointly learn spatial and temporal features from videos. However, their large computational complexities impede them from being applied in real scenarios. In contrast, 2D CNNs are light-weight, but do not bear the ability for temporal modeling. To bridge the gap between image recognition and video recognition, considerable 2D-CNN-based researches [23, 9, 10, 33, 34] recently attempt to equip the conventional 2D CNNs with a temporal modeling ability, and some improvements are witnessed.

However, another direction seems to be less explored for 2D-CNN-based video analysis, namely diversifying visual representations among video frames. Although the 2D CNN takes multiple frames of a video at once as input, the frames captured from the same scene share the same convolution kernels. A fact about CNN is that each feature channel generated by the kernel convolution from the high-level layers highly reacts to a specific semantic pattern. Hence, with 2D CNN, the yielded features from different frames may share multiple similar channels, which thereafter results in repeated and redundant information extraction for video analysis. If the majority part of frames is background, these repeated redundant channels tend to describe the background scene rather than the regions of interest. This tendency may lead to the ignorance of the motion information which can be more critical than the scene information for action understanding [8, 24, 27, 2]. Besides, the customary strategy that features from different frames of a video are learned under the same label of supervision will make this issue even more severe. We observe that for one temporal reasoning dataset like Something-Something [5], video samples under the same category are from various scenes and the actions therein are performed with various objects. The scene and object information may not be directly useful for the recognition task. Thus, a 2D-CNN-based method like TSN [23] is easy to overfit as the model learns many scene features and meanwhile neglects the variations among frames, *e.g.* the motion information. We state that due to this redundant information extraction, the previously proposed temporal modeling method cannot fully play its role. In this paper, we propose two ways to tackle the issue.

We first introduce an information filtering module, i.e., Progressive Enhancement Module (PEM), to adaptively and sequentially enhance the discriminative channels and meanwhile suppress the repeated ones of each frame's feature with the help of motion historical information. Specifically, the PEM progressively determines the enhancements for the current frame's feature maps based on the motion observation in previous time steps. This sequential way of enhancement learning explicitly takes the temporal order of frames into consideration, which enables the network itself to effectively avoid gathering similar channels and fully utilize the information from different temporal frames. After PEM, we set

a temporal modeling module that temporally fuses the enhanced features to help the discriminative information from different frames interact with each other.

Furthermore, the convolution kernels are calibrated by the Temporal Diversity Loss (TD Loss) so that they are forced to concentrate on and capture the variations among frames. We locate a loss right after the temporal modeling module. By minimizing the pair-wise cosine similarity of the same channels between different frames, the kernels can be well adjusted to diversify the representations across frames. As the TD Loss acts as a regularization enforced to the network training, it does not add an extra complexity to the model and keeps a decent accuracy-speed tradeoff.

We evaluate our method on three benchmark datasets. The proposed model outperforms the best state-of-the-arts by 2.4%, 1.3% and 0.8% under the $8f$ setting on the Something-SomethingV1, V2 and the Kinetics400 datasets, respectively, as shown in Table 1 and Table 2. The proposed PEM and TD Loss outperform the baseline by 2.6% and 2.3% on Something-Something V1, respectively. The experimental results demonstrate the effectiveness of our proposed 2D-CNN-based model on video analysis.

Our contributions can be summarized as follows:

- We propose a Progressive Enhancement Module for channel-level information filtering, which effectively excites the discriminative channels of different frames and meanwhile avoids repeated information extraction.

- We propose a Temporal Diversity Loss to train the network. The loss calibrates the convolution kernels so that the network can concentrate on and capture the variations among frames. The loss also improves the recognition accuracy without adding an extra network complexity.

2 Related Work

2D-CNNs for Video Analysis Due to the previous great success in classifying images of objects, scenes, and complex events [3, 18, 19, 6], convolutional neural networks have been introduced to solve the problem of video understanding. Using two-dimensional convolutional network is a straightforward way to characterize videos. In Temporal Segment Network [23], 2D CNN is utilized to individually extract a visual representation for each sampled frame of a video, and an average pooling aggregation scheme is applied for temporal modeling. To further tackle the temporal reasoning of 2D CNNs, Zhou *et.al.* proposed a Temporal Relational Network [33] to hierarchically construct the temporal relationship among video frames. Ji *et.al.* introduced a simple but effective shift operation between frames into 2D CNN, and proposed the Temporal Shift Module [10]. Following the same direction, the Temporal Enhancement Interaction Network (TEINet) [11] introduces a depth-wise temporal convolution for light-weight temporal modeling. Similar methods include Temporal Bilinear Network [9] and Approximate Bilinear Module [34], which re-design the bilinear pooling for temporal modeling.

These methods attempt to equip the 2D CNN with an ability of temporal modeling. However, one neglected limitation of the 2D CNN is the redundant feature extraction among frames or the lack of temporal representation diversity. This is the battle field to which our proposed method is engaged. We first propose a Progressive Enhancement Module before temporal modeling to enhance the discriminative channels and meanwhile suppress the redundant channels of different frames sequentially. Furthermore, after the temporal modeling module, we create a Temporal Diversity loss to force the convolution kernels to capture the variations among frames.

Channel Enhancement The idea of enhancing the discriminative channels for recognition first appears in image recognition. In Squeeze-and-Excitation Network (SENet) [7], an attention sub-branch in the convolutional layer is involved to excite the discriminative channels of frame's features. Inheriting from the SENet, to emphasize the motion cues in videos, TEINet uses the difference between feature maps of two consecutive frames for channel-level enhancement learning. In our method, we expand the receptive field of this channel enhancement module. At each time step, the enhancement module is able to be aware of the motion conducted in previous frames, therefore avoiding activating the channels emphasized previously.

Diversity Regularization In fine-grained image recognition, to adaptively localize discriminative parts, attention models are widely used. However, the previously proposed attention models perform poorly in classifying fine-grained objects as the learned attentions tend to be similar to each other. In [32, 31], attention maps are regularized to be diverse in the spatial domain to capture the discriminative parts. In this paper, we take the temporal diversity of the feature maps into consideration and propose the Temporal Diversity Loss. The TD Loss directly sets the regularization on the visual representation of each frame to obtain the discriminative and dynamic features for video analysis.

3 Proposed Method

In this section, we elaborate on the two contributions of this work. We first give the framework of the proposed method in Sec. 3.1. The Progressive Enhancement Module is introduced in Sec. 3.2. In Sec. 3.3, the Temporal Diversity Loss for diverse representation modeling is described. The illustration of the whole framework is shown in Fig. 1.

3.1 Framework

In our model, each video frame is represented by a tensor $\boldsymbol{X}_t^b \in \mathbb{R}^{C_b \times W_b \times H_b}$, which bears C_b stacked feature maps with width W_b and height H_b, and b indicates the block index. In the following, we use C, W and H instead to simplify the

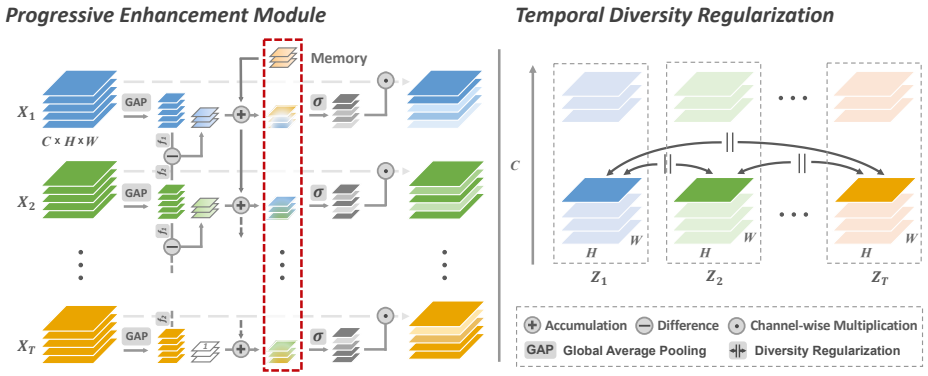

Fig. 1: An illustration of the proposed framework. In PEM, 1) features of each frame are GAP-ed and down-sampled to get a vector. 2) The differencing operation is performed on the vectors of each two consecutive frames. 3) The *memory* vector (in the red box) accumulates historical difference information. 4) With the Sigmoid function, the channel-level enhancement is obtained to excite discriminative channels of each frame. To compress the model complexity, the 1×1 convolution operation in $f(\cdot)$ reduces the vector dimensionality and the one before $\sigma(\cdot)$ recovers it back to C. In TD regularization, the same channels of each frame pair are regularized to be distinguished from each other.

notations. Given a video sequence/clip with T sampled frames $V = \{\boldsymbol{X}_t\}_{t=1}^{T}$, the goal of our established deep network is to extract a discriminative visual representation of V and predict its class label $k \in \{1, 2, ..., K\}$, where K is the number of categories. Each block of the network takes the T frames as input and outputs the feature maps for the next block, which is formulated as follows:

$$(\boldsymbol{X}_1^b, \cdots, \boldsymbol{X}_T^b) = \mathcal{F}(\boldsymbol{X}_1^{b-1}, \cdots, \boldsymbol{X}_T^{b-1}; \boldsymbol{\theta}^b), \tag{1}$$

where $\mathcal{F}$ is the feature mapping function, which involves the Progressive Attention for information filtering (Sec. 3.2) and temporal information interaction. $\boldsymbol{\theta}$ is the block parameters to be optimized. The input and output of the network are denoted as $\boldsymbol{X}^0$ and $\boldsymbol{X}^B$, and B is the total number of blocks.

The output feature maps $\{\boldsymbol{X}_t^B\}_{t=1}^{T}$ are gathered by average pooling, and are further fed into the Softmax function for category prediction. This mapping is defined as $\hat{\boldsymbol{y}} = \mathcal{G}(\boldsymbol{X}_1^B, \cdots, \boldsymbol{X}_T^B)$, where $\hat{\boldsymbol{y}} \in [0, 1]^K$ contains the prediction scores of K categories. Therefore, the loss function is defined as

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_r = -\sum_{i=1}^{K} y_i \cdot log\, \hat{y}_i + \lambda \mathcal{L}_r, \tag{2}$$

where $\mathcal{L}_c$ is the Cross Entropy Loss for category supervision. y_i is the groundtruth label concerning class i, and it is an element of the one-hot vector $\boldsymbol{y} \in \{0, 1\}^K$. $\mathcal{L}_r$ is the regularization term for network training, and λ balances the importance

between category supervision and network regularization. To enhance the temporal diversity of feature maps from different frames and thereafter to model the crucial motion information, the regularization term is defined as the Temporal Diversity Loss as depicted in Sec. 3.3.

3.2 Progressive Enhancement Module

As discussed in Sec. 1, one drawback of using 2D CNN for video analysis is that most of the kernels in one convolutional network are inclined to focus on repeated information, like scenes, across the features from different time steps, which cannot easily take full advantage of information from the video. The Progressive Enhancement Module (PEM) can sequentially determine which channels of each frame's features to focus on, and therefore effectively extract action related information. In each block, the feature maps $\{\boldsymbol{X}_t^{b-1}\}_{t=1}^T$ from the preceding block are first fed into the Progressive Enhancement Module for information filtering, as illustrated in Fig. 1. Let $\boldsymbol{a}_t^b \in \mathbb{R}^C$ denote the enhancement vector to excite the discriminative channels of each frame. This operation is defined as

$$\boldsymbol{U}_t^b = \boldsymbol{X}_t^{b-1} \odot \boldsymbol{a}_t^b \,, \tag{3}$$

where $\boldsymbol{U}_t^b$ is the t-th frame output of PEM in the b-th block, and $\odot$ is a channel-wise multiplication. For notational simplicity, we remove the block-index notation b in the following description.

The input feature maps $\{\boldsymbol{X}_t^{b-1}\}_{t=1}^T$ are first aggregated across the spatial dimensions by using Global Average Pooling (GAP), and the channel-wise statistics $\{\boldsymbol{x}_t\}_{t=1}^T$, $\boldsymbol{x} \in \mathbb{R}^C$ are then obtained. Each pair of neighboring frames in $\{\boldsymbol{x}_t\}_{t=1}^T$ is then fed into two individual 1×1 convolution operations f_1 and f_2 with ReLU activation, respectively, for feature selection. As discussed in [11], taking the difference of channel statistics between two consecutive frames as input for channel-level enhancement learning is more effective for video analysis than the original channel-wise statistics $\{\boldsymbol{x}_t\}_{t=1}^T$ proposed in Squeeze-and-Excitation Network [7], which is especially designed for image recognition. We choose to use the difference of channel statistics between two consecutive frames as the input of PEM. With the differencing operation, we obtain the difference of channel-wise statistics $\{\boldsymbol{d}_t\}_{t=1}^T$. The differencing operation is defined as

$$\boldsymbol{d}_t = f_2(\boldsymbol{x}_{t+1}) - f_1(\boldsymbol{x}_t) \,, \tag{4}$$

and the difference of the last frame, $\boldsymbol{d}_T$, is set as a vector with ones to maintain the magnitude of the memory vector .

To extend the receptive field of enhancement learning, we here introduce an accumulation operation into the learning of channel-level enhancement for each frame. By the accumulation, the enhancement module of each current frame can be aware of the vital motion information in the previous timings, and not be trapped into the local temporal window as in [11]. The accumulated vector $\boldsymbol{m}$, named as *memory*, accumulates $\boldsymbol{d}$ at each time step, and the accumulation operation is controlled by $\gamma \in [0, 1]$, as defined in Eq. (5):

$$\boldsymbol{m}_t = (1-\gamma)\cdot \boldsymbol{m}_{t-1} + \gamma \cdot \boldsymbol{d}_t\,, \qquad \gamma = \sigma(\boldsymbol{W}_g(\boldsymbol{m}_{t-1} \parallel \boldsymbol{d}_t))\,. \tag{5}$$

The factor γ is determined by the accumulated vector $\boldsymbol{m}_{t-1}$ and the difference information $\boldsymbol{d}_t$, where $\parallel$ denotes a concatenation operation, $\boldsymbol{W}_g$ is a projection matrix for linear transformation, and $\sigma(\cdot)$ is a Sigmoid activation function. The final enhancement vector $\boldsymbol{a}$ is then generated by

$$\boldsymbol{a}_t = \sigma(\boldsymbol{W}_a \boldsymbol{m}_t)\,, \tag{6}$$

where $\boldsymbol{W}_a$ is a matrix linearly projecting $\boldsymbol{m}$ into a new vector space. With PEM, the network is able to progressively select the motion-related channels in each frame, and adaptively filter the discriminative information for video analysis. The enhanced feature maps $\{\boldsymbol{U}_t\}_{t=1}^{T}$ are then fed into a temporal modeling module for temporal information fusion, and we write the output as $\{\boldsymbol{Z}_t\}_{t=1}^{T}$, $\boldsymbol{Z} \in \mathbb{R}^{C\times W\times H}$.

3.3 Temporal Diversity Loss

It is well-known that feature maps from high-level layers tend to have responses to specific semantic patterns. Convolution kernels that focus on the background of a video may generate similar semantic patterns for the same channels of features from different frames, which may lead to redundant visual feature extraction for video analysis. To calibrate the kernels in 2D CNN and force the network to focus on and capture the variations among frames of a video sequence, we propose the Temporal Diversity Loss to regularize the network toward learning distinguished visual features for different frames. For the feature map $\boldsymbol{Z}_t$ from each frame, its C vectorized channel features are denoted as $\{\boldsymbol{z}_t^c\}_{c=1}^{C}$, $\boldsymbol{z} \in \mathbb{R}^{WH}$. We use the Cosine Similarity to measure the similarities of a specific channel between two frames of each video frame pair, and then define the loss as:

$$\mathcal{L}_\mu = \sum_c \frac{1}{|\mathbb{I}|} \sum_{(i,j)\in\mathbb{I}} \eta(\boldsymbol{z}_i^c, \boldsymbol{z}_j^c)\,, \tag{7}$$

where $\mathbb{I} = \{(i,j)\,|\,i \neq j,\, 1 \leqslant i,j \leqslant T\}$, $|\cdot|$ indicates the total number of elements in a set, and $\eta(\cdot)$ defines the Cosine Similarity measure, namely $\eta(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}^\intercal \boldsymbol{y}}{\|\boldsymbol{x}\|_2 \cdot \|\boldsymbol{y}\|_2}$. Considering that the static information among frames is also beneficial to recognition, we only use C_μ ($C_\mu < C$) channels for temporal diversity regularization. A further analysis will be discussed in Sec. 4.5. With the proposed Temporal Diversity $\mathcal{L}_\mu$, the regularization term $\mathcal{L}_r$ is then defined as $\mathcal{L}_r = \sum_{b=1}^{B_\mu} \mathcal{L}_\mu^b$, where B_μ is the number of blocks with temporal diversity regularization.

4 Experiments

In this section, the proposed method is evaluated on three benchmark datasets, the Something-Something V1 dataset [5], Something-Something V2 dataset [16], and the Kinetics400 dataset [1]. We first briefly introduce these three datasets

and the experiment settings in Sec. 4.1 and Sec. 4.2, respectively. Then, our method is compared with the state-of-the-arts in Sec. 4.3. The ablation study is conducted in Sec. 4.4 to evaluate the performance of each individual module of our proposed method. In Sec. 4.5, we evaluate the proposed method in detail, including parameter, position sensitivity analysis and visualization.

4.1 Datasets

Something-Something V1&V2 are crowd-sourced datasets focusing on temporal modeling. In these two datasets, the scenes and objects in each single action category are various, which strongly requires the considered model to focus on the temporal variations among video frames. The V1 & V2 datasets include 108,499/220,847 videos, respectively, containing 174 action categories in both versions.

Kinetics400 is a large-scale YouTube-like dataset, which contains 400 human action classes, with at least 400 video clips for each category. The average duration of video clips in this dataset is around 10s. Unlike Something-Something datasets, Kinetics is less sensitive to temporal relationships, so the scene information is of importance in its recognition.

4.2 Experimental Setup

In all the conducted experiments, we use the ResNet-50 [6] as our backbone considering the tradeoff between performance and efficiency, and our model is pre-trained by ImageNet [3]. We set the Progressive Enhancement Module (PEM) in front of all the blocks of the ResNet backbone. Given that the early stages of the Convolutional Network focus more on spatial appearance modeling and the later ones focus on temporal modeling [21, 15] and for better convergence, we regularize the temporal diversity of feature maps in the last blocks of each of the last three layers. The Temporal Diversity Loss (TD Loss) is located right after the temporal modeling module. What follows the temporal modeling module is the convolution operation (ResConv) taken from the ResNet block, which includes one 1×1, one 3×3, and one 1×1 2D convolutions. The position of the PEM and the temporal diversity regularization are illustrated in Fig. 2. We define the TDRL-A as the block without the TD Loss, and TDRL-B as the one with the TD Loss, where TDRL stands for Temporal Distinct Representation Learning. The ratio of channels regularized by Temporal Diversity Loss in each feature is 50%. Without loss of generality, we

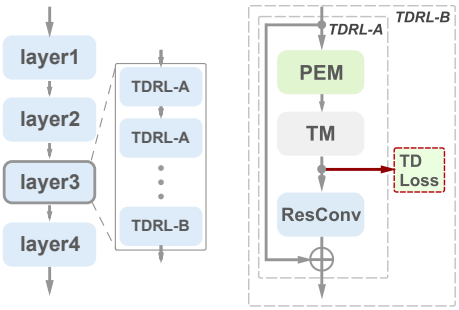

Fig. 2: Block Illustration

Table 1: Comparison with the state-of-the-arts on Something-Something V1&V2 (Top1 Accuracy %). The notation 'I' or 'K' in the backbone column indicates that the model is pre-trained with ImageNet or Kinetics400 dataset. The subscripts of 'Val' and 'Test' indicate the version of the Something-Something dataset. '2S' indicates two streams.

Method	**Backbone**	**Frames**	**FLOPs**	**Val$_1$**	**Test$_1$**	**Val$_2$**	**Test$_2$**
I3D [26]	Res3D-50 (IK)	$32f \times 2$	306G	41.6	–	–	–
NL I3D [26]			334G	44.4	–	–	–
NL I3D+GCN [26]			606G	46.1	45.0	–	–
TSM [10]	Res2D-50 (IK)	$8f$	33G	45.6	–	–	–
		$8f \times 2$	65G	47.3	–	61.7	–
		$16f$	65G	47.2	46.0	–	–
TSM$_{En}$ [10]		$16f + 8f$	98G	49.7	–	–	–
TSM-2S [10]		$16f + 16f$	–	52.6	50.7	64.0	64.3
TEINet [11]	Res2D-50(I)	$8f$	33G	47.4	–	61.3	60.6
		$8f \times 10$	990G	48.8	–	64.0	62.7
		$16f$	66G	49.9	–	62.1	60.8
		$16f \times 10$	1980G	51.0	44.7	64.7	63.0
TEINet$_{En}$ [11]		$8f + 16f$	99G	52.5	46.1	66.5	64.6
GST [15]	Res2D-50(I)	$8f$	29.5G	47.0	–	61.6	60.0
		$16f$	59G	48.6	–	62.6	61.2
Ours	Res2D-50(I)	$8f$	33G	**49.8**	**42.7**	**62.6**	**61.4**
		$8f \times 2$	198G	**50.4**	–	**63.5**	–
		$16f$	66G	**50.9**	**44.7**	**63.8**	**62.5**
		$16f \times 2$	396G	**52.0**	–	**65.0**	–
		$8f + 16f$	99G	**54.3**	**48.3**	**67.0**	**65.1**

use the Temporal Interaction Module proposed in [11] as our temporal modeling module (TM). λ for loss balancing is set as 2×10^{-4}.

Pre-processing We follow a similar pre-processing strategy to that described in [25]. To be specific, we first resize the shorter side of RGB images to 256, and center crop a patch followed by scale-jittering. The image patches are then resized to 224×224 before being fed into the network. Owing to the various lengths of video sequences, we adopt different temporal sampling strategies for different datasets. The network takes a clip of a video as input. Each clip consists of 8 or 16 frames. For the Kinetics dataset, we uniformly sample 8 or 16 frames from the consecutive 64 frames randomly sampled in each video. For the Something-Something dataset, due to the limited duration of video samples, we uniformly sample 8 or 16 frames from the whole video.

Training For the Kinetics dataset, we train our model for 100 epochs. The initial learning rate is set as 0.01, and is scaled with 0.1 at 50, 75, and 90 epochs. For

Table 2: Comparison with the state-of-the-arts on Kinetics400 (%). The notations 'I', 'Z', 'S' in the backbone column indicate that the model is pre-trained with ImageNet, trained from scratch, or pre-trained with the Sport1M dataset, respectively.

Method	Backbone	GFLOPs×views	Top-1	Top-5
$I3D_{64f}$ [26]	Inception V1(I)	108×N/A	72.1	90.3
$I3D_{64f}$ [26]	Inception V1(Z)	108×N/A	67.5	87.2
NL+$I3D_{32f}$ [25]	Res3D-50(I)	70.5×30	74.9	91.6
NL+$I3D_{128f}$ [25]	Res3D-50(I)	282×30	76.5	92.6
NL+$I3D_{128f}$ [25]	Res3D-101(I)	359×30	77.7	93.3
Slowfast [4]	Res3D-50(Z)	36.1×30	75.6	92.1
Slowfast [4]	Res3D-101(Z)	106×30	77.9	93.2
NL+Slowfast [4]	Res3D-101(Z)	234×30	**79.8**	**93.9**
LGD-$3D_{128f}$ [17]	Res3D-101(I)	N/A×N/A	79.4	94.4
$R(2+1)D_{32f}$ [21]	Res2D-34(Z)	152×10	72.0	90.0
$R(2+1)D_{32f}$ [21]	Res2D-34(S)	152×10	74.3	91.4
$ARTNet_{16f}$+TSN [22]	Res2D-18(Z)	23.5×250	70.7	89.3
S3D-G_{64f} [28]	Inception V1(I)	71.4×30	74.7	93.4
TSM_{16f} [10]	Res2D-50(I)	65×30	74.7	91.4
$TEINet_{8f}$ [11]	Res2D-50(I)	33×30	74.9	91.8
$TEINet_{16f}$ [11]	Res2D-50(I)	66×30	76.2	92.5
$Ours_{8f}$	Res2D-50(I)	33×30	**75.7**	**92.2**
$Ours_{16f}$		66×30	**76.9**	**93.0**

the Something-Something dataset, the model is trained with 50 epochs in total. The initial learning rate is set as 0.01 and reduced by a factor of 10 at 30, 40, and 45 epochs. In the training, Stochastic Gradient Decent (SGD) is utilized with momentum 0.9 and weight decay of 1×10^{-4}. The experiments are conducted on *Tesla M40* GPUs, and the batch size is set as 64. The memory vector can be initialized by zeros, or the difference vector $\boldsymbol{d}_t$ from the first or last frame, and we experimentally find that the last frame difference $\boldsymbol{d}_{T-1}$ can achieve slightly better performance. We therefore use $\boldsymbol{d}_{T-1}$ as the memory initialization in the experiments.

Inference For fair comparison with the state-of-the-arts, we follow two different data processing settings to evaluate our model. In single-clip (8 or 16 frames) comparison, namely model trained with 8 frames only ($8f$), or with 16 frames only ($16f$), we use center cropping for input image processing. The analysis experiments are under the $8f$ setting. In multi-clip comparison, we follow the widely applied settings in [25, 10] to resize the shorter side of images to 256 and take 3 crops (left, middle, right) in each frame. Then we uniformly sample N clips ($8f \times N$ or $16f \times N$) in each video and obtain the classification scores for each clip individually, and the final prediction is based on the average classification score of the N clips.

Table 3: Ablation Study - TSM [10] (%)

Method	Top-1	Top-5
baseline [10]	45.6	74.2
+PEM	48.1	77.4
+TDLoss	47.5	76.8
+PEM+TDLoss	48.4	77.4

Table 4: Ablation Study - TIM [11] (%)

Method	Top-1	Top-5
baseline [11]	46.1	74.7
+MEM [11]	47.4	76.6
+PEM	48.7	77.8
+TDLoss	48.4	77.3
+PEM+TDLoss	49.8	78.1

4.3 Comparison with State-of-the-Arts

We compare the proposed method with the state-of-the-arts on the Something-SomethingV1&V2 and the Kinetics400 datasets under different settings for fair comparison. The results are shown in Table 1 and Table 2, respectively. Table 1 shows that on the Something-Something V1 dataset, our proposed method outperforms the so far best model, TEINet [11], by 2.4%, 1.3%, and 1.8% under the $8f$, $16f$, and $8f + 16f$ settings on the validation set, respectively. Our performance under the two-clips setting is even better than TEINet's performance under the ten-clips setting. On the Something-Something V2 dataset, the performance of our model under the $8f$ setting is even better or the same as the TEINet and GST under the $16f$ setting, which indicates that we can use only half of the inputs of these two models to achieve the same or better accuracy. These results verify the effectiveness of the temporal representation diversity learning. On the Kinetics dataset, the results are reported under the ten-clips-three-crops setting. As can be seen from Table 2, our proposed model outperforms all the 2D-CNN-based models under different settings, and it even performs better than the 3D-CNN-based nonlocal [25] and slowfast [4] networks with less frames input. We can also witness consistent improvement on the test set of V2, and our model beats TEI by 1.2% under the $8f + 16f$ setting.

4.4 Ablation Study

In this section, we evaluate the performances of different modules in our model. We use the single-clip $8f$ setting for the experiment conducted in this section. We use a temporal shift module (TSM) [10] and a temporal interaction module (TIM) [11], respectively, as the baseline in our experiment to show the generality of our model cooperating with different temporal modeling modules. As can be seen from Table 3, with the PEM and the TD Loss, there are 2.5% and 1.9% Top-1 accuracy improvements over the TSM, respectively. With both the PEM and TD Loss, the improvement reaches 2.8%. Similarly, as shown in Table 4, PEM gives 2.6% Top-1 accuracy improvement over the baseline, and it outperforms MEM [11] by 1.3%, which also involves a channel enhancement module. With the TD Loss, there is 2.3% improvement over the baseline. We can see from Table 4 that there is 3.7% improvement over the baseline when both the PEM and TD Loss are applied. One more thing we need to point out is that, after 50 epochs

Table 5: TDLoss Ratio (%)

Method	Top-1
baseline	46.1
+25% TDLoss	47.7
+50% TDLoss	48.4
+75% TDLoss	47.8
+100% TDLoss	47.3

Table 6: Block Position (%)

Order	Top-1
TM	46.1
PEM B. TM	48.7
PEM A. TM	49.0
TDLoss B. TM	46.9
TDLoss A. TM	48.4

Table 7: Impact of λ (%)

λ	Top-1
0×10^{-4}	46.1
1×10^{-4}	47.9
2×10^{-4}	48.4
3×10^{-4}	47.8
4×10^{-4}	48.0

training, the TIM baseline's training accuracy reaches 79.03%, while with the PEM and the TD Loss, the training accuracies are down to 77.98% and 74.45%, respectively. This training accuracy decline shows that our proposed method can avoid overfitting, and force the model to learn the essential motion cues.

4.5 Detailed Analysis

Ratio of Channel Regularization The ratio of channel regularization indicates that how many channels are involved for diversity regularization. We set the ratio from 0% (baseline) to 100% to evaluate the impact of the TD loss in this section. The results are shown in Table 5. From this table we can see that when half of the channels are regularized, the model achieves the best performance, 48.4%. If all the channels are set under the regularization, the model reaches the lowest accuracy 47.3%. This comparison shows that not all the channels require the regularization, and channels without regularization are still useful for the recognition task. However, no matter how many channels are involved in the loss, good improvement is still witnessed over the baseline, TIM.

Position of Blocks In this part, we discuss where to insert the PEM and TD Loss in each block. The two modules are located before (B.) or after (A.) the temporal module (TM) individually to see the impact of position on accuracy. We follow the position of MEM in TEINet [11] for fair comparison. The results are shown in Table 6. As can be seen, for PEM, there is no much difference between the two locations. It can fairly provide stable improvement on two different positions. For the TD loss, we discover that when it is located before the TM, the improvement over the baseline is limited. Because TM is inclined to make the representation similar to each other, the following ResConv cannot well extract the video representation. While when the TD regularization is after TM, there is 2.3% improvement over the baseline. The TD loss effectively diversifies the representations among different frames after TM.

Loss Balancing Factor λ We analyze the impact of the loss balancing factor λ on the accuracy in this section. We set λ from 1×10^{-4} to 4×10^{-4}. The result comparisons are shown in Table 7. As can be seen, the fluctuation is in the range of 0.6%, which shows that the proposed TD Loss is not very sensitive

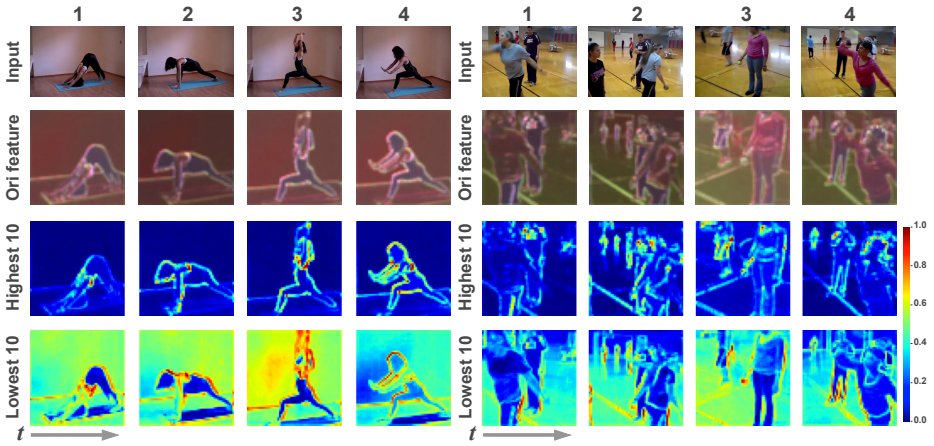

Fig. 3: Visualization of the features excited and suppressed by PEM. The features are organized in the temporal order, and they are from the first block of the first layer. **R1**: the images in this row are the input images of the network. **R2**: the features in this row are those before PEM. The channels of each of these features are divided by three groups, and the channels in each group are gathered by average pooling to generate a 3-channels feature, presented as an RGB image. **R3-4**: Each of the feature map in the these two rows is the average of channels picked from the features before PEM. Each feature map in the third row is gathering of ten channels with the highest enhancement, and each one in the fourth row is gathered with the lowest enhancement.

to λ when this factor is in an appropriate range. No matter which λ we set, the involvement of the TD Loss can still help improve the accuracy over the baseline, which shows the effectiveness of the proposed temporal diversity loss.

Visualization We visualize feature maps from different blocks to show the effect of the PEM and TD Loss. The experiment is conducted under the Kinetics dataset. We show the feature maps filtered by the PEM in Fig. 3. There are two video samples shown in the figure. The input images are uniformly sampled from a video. From Fig. 3 we can see that the top ten enhanced channels mainly focus on the motion, while the top ten suppressed channels highly respond to the static background. This visualization shows that the proposed PEM can well discover which are the motion-related channels and which are the repeated static background channels. By enhancing the motion-related ones and suppress the repeated ones, the redundant information can be filtered out and the discriminative one can be well kept. As can be seen from Fig. 4, with the TD Loss, the feature maps after TM can well encode the information from the current frame and its neighboring frames, while the motion encoded in the features after TM without the TD regularization is very limited. The figures indicate that the TD loss can calibrate the temporal convolution kernels and also enhance the temporal interactions among them.

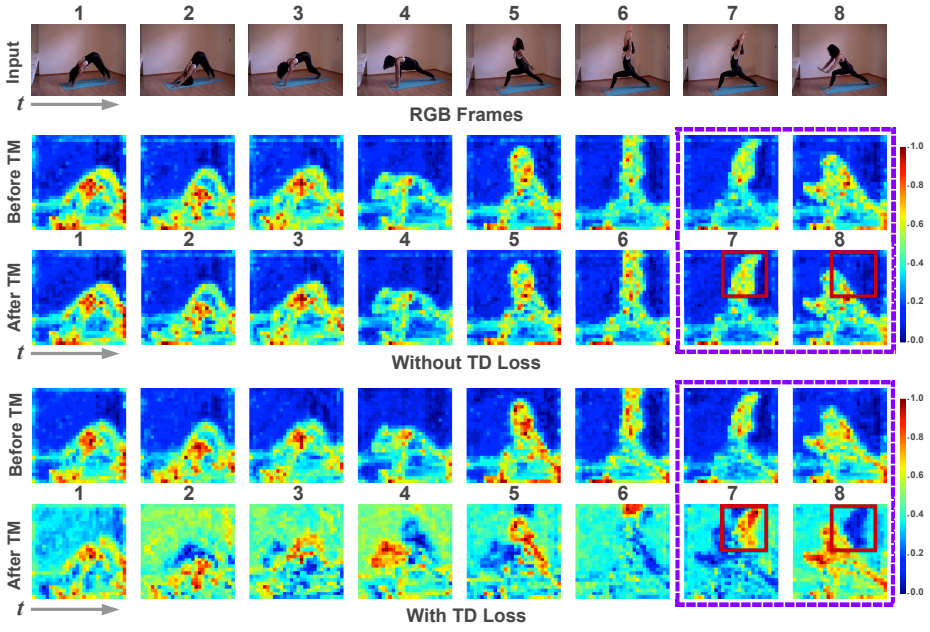

Fig. 4: Visualization of feature maps before and after TM w/ or w/o the TD loss. These feature maps are uniformly sampled from one video and are organized following the temporal order. They are from the last block of the second layer. **R1**: The images are the input to the network. The purple dashed rectangles mark and illustrate the difference between feature maps with and without TD Loss. **w/ TD Loss**, the feature maps can well encode action from neighboring frames, and emphasize the variations among them, as marked by red rectangles in the last row. **w/o TD loss**, the features cannot enhance those variations, as marked by red rectangles in the third row.

5 Conclusions

In this work, we proposed two ways to tackle the issue that the 2D CNN cannot well capture large variations among frames of videos. We first introduced the Progressive Enhancement Module to sequentially excite the discriminative channels of frames. The learned enhancement can be aware of the frame variations in the past time and effectively avoid the redundant feature extraction process. Furthermore, the Temporal Diversity Loss was proposed to diversify the representations after temporal modeling. With this loss, the convolutional kernels are effectively calibrated to capture the variations among frames. The experiments were conducted on three datasets to validate our contributions, showing the effectiveness of the proposed PEM and TD loss.

Acknowledgement. We thank Dr. Wei Liu from Tencent AI Lab for his valuable advice.

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 6299–6308 (2017)
2. Choi, J., Gao, C., Messou, J.C., Huang, J.B.: Why can't i dance in the mall, learning to mitigate scene bias in action recognition. In: NeurIPS. pp. 853–865 (2019)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. Ieee (2009)
4. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV. pp. 6202–6211 (2019)
5. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: ICCV. vol. 1, p. 5 (2017)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
7. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR. pp. 7132–7141 (2018)
8. Jiang, Y.G., Dai, Q., Xue, X., Liu, W., Ngo, C.W.: Trajectory-based modeling of human actions with motion reference points. In: ECCV. pp. 425–438. Springer (2012)
9. Li, Y., Song, S., Li, Y., Liu, J.: Temporal bilinear networks for video action recognition. In: AAAI. vol. 33, pp. 8674–8681 (2019)
10. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: ICCV. pp. 7083–7093 (2019)
11. Liu, Z., Luo, D., Wang, Y., Wang, L., Tai, Y., Wang, C., Li, J., Huang, F., Lu, T.: Teinet: Towards an efficient architecture for video recognition. In: AAAI. vol. 2, p. 8 (2020)
12. Lu, X., Ma, C., Ni, B., Yang, X., Reid, I., Yang, M.H.: Deep regression tracking with shrinkage loss. In: ECCV. pp. 353–369 (2018)
13. Lu, X., Wang, W., Ma, C., Shen, J., Shao, L., Porikli, F.: See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In: CVPR. pp. 3623–3632 (2019)
14. Lu, X., Wang, W., Shen, J., Tai, Y.W., Crandall, D.J., Hoi, S.C.: Learning video object segmentation from unlabeled videos. In: CVPR. pp. 8960–8970 (2020)
15. Luo, C., Yuille, A.L.: Grouped spatial-temporal aggregation for efficient action recognition. In: ICCV. pp. 5512–5521 (2019)
16. Mahdisoltani, F., Berger, G., Gharbieh, W., Fleet, D., Memisevic, R.: On the effectiveness of task granularity for transfer learning. arXiv:1804.09235 (2018)
17. Qiu, Z., Yao, T., Ngo, C.W., Tian, X., Mei, T.: Learning spatio-temporal representation with local and global diffusion. In: CVPR. pp. 12056–12065 (2019)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2014)
19. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. pp. 1–9 (2015)
20. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. pp. 4489–4497 (2015)

21. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR. pp. 6450–6459 (2018)
22. Wang, L., Li, W., Li, W., Van Gool, L.: Appearance-and-relation networks for video classification. In: CVPR. pp. 1430–1439 (2018)
23. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV. pp. 20–36. Springer (2016)
24. Wang, X., Farhadi, A., Gupta, A.: Actions transformations. In: CVPR. pp. 2658–2667 (2016)
25. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR. pp. 7794–7803 (2018)
26. Wang, X., Gupta, A.: Videos as space-time region graphs. In: ECCV. pp. 399–417 (2018)
27. Wang, Y., Hoai, M.: Pulling actions out of context, explicit separation for effective combination. In: CVPR. pp. 7044–7053 (2018)
28. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: ECCV. pp. 305–321 (2018)
29. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: NeurIPS. pp. 802–810 (2015)
30. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: ICCV. pp. 2117–2126 (2017)
31. Zhao, B., Wu, X., Feng, J., Peng, Q., Yan, S.: Diversified visual attention networks for fine-grained object classification. T-MM **19**(6), 1245–1256 (2017)
32. Zheng, H., Fu, J., Mei, T., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: CVPR. pp. 5209–5217 (2017)
33. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: ECCV. pp. 803–818 (2018)
34. Zhu, X., Xu, C., Hui, L., Lu, C., Tao, D.: Approximated bilinear modules for temporal modeling. In: ICCV. pp. 3494–3503 (2019)