TDN: Temporal Difference Networks for Efficient Action Recognition

Limin Wang Zhan Tong Bin Ji Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University, China
07wanglimin@gmail.com, tongzhan@smail.nju.edu.cn, binjinju@smail.nju.edu.cn, gswu@nju.edu.cn

Abstract

Temporal modeling still remains challenging for action recognition in videos. To mitigate this issue, this paper presents a new video architecture, termed as Temporal Difference Network (TDN), with a focus on capturing multi-scale temporal information for efficient action recognition. The core of our TDN is to devise an efficient temporal module (TDM) by explicitly leveraging a temporal difference operator, and systematically assess its effect on short-term and long-term motion modeling. To fully capture temporal information over the entire video, our TDN is established with a two-level difference modeling paradigm. Specifically, for local motion modeling, temporal difference over consecutive frames is used to supply 2D CNNs with finer motion pattern, while for global motion modeling, temporal difference across segments is incorporated to capture long-range structure for motion feature excitation. TDN provides a simple and principled temporal modeling framework and could be instantiated with the existing CNNs at a small extra computational cost. Our TDN presents a new state of the art on the Something-Something V1 & V2 datasets and is on par with the best performance on the Kinetics-400 dataset. In addition, we conduct in-depth ablation studies and plot the visualization results of our TDN, hopefully providing insightful analysis on temporal difference modeling. We release the code at https://github.com/MCG-NJU/TDN.

1. Introduction

Deep neural networks have witnessed great progress for action recognition in videos [14, 29, 38, 31, 6, 26, 37]. Temporal modeling is crucial for capturing motion information in videos for action recognition, and this is usually achieved by two kinds of mechanisms in the current deep learning approaches. One common method is to use a two-stream network [29], where one stream is on RGB frames to extract appearance information, and the other is to leverage optical flow as an input to capture movement information. This method turns out to be effective for improving action recognition accuracy but requires high computational consumption for optical flow calculation. Another alternative approach is to use 3D convolutions [12, 31] or temporal convolutions [33, 41, 25] to implicitly learn motion features from RGB frames. However, 3D convolutions often lack specific consideration in the temporal dimension and might bring higher computational cost as well. Therefore, designing an effective temporal module of high motion modeling power and low computational consumption is still a challenging problem for video recognition.

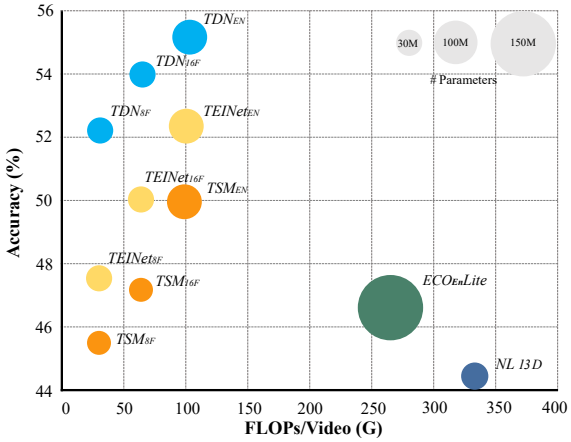

Figure 1. Video classification performance comparison on Something-Something V1 [8] in terms of Top1 accuracy, computational cost, and model size. Our proposed TDN achieves the best trade-off between accuracy and efficiency, when compared with previous methods such as NL I3D [40], ECO [46], TSM [19] and TEINet [20].

This paper aims to present a new temporal modeling mechanism by introducing a temporal difference based module (TDM). Temporal derivative (difference) is highly relevant with optical flow [11], and has shown effectiveness in action recognition by using RGB difference as an approximate motion representation [38, 43]. However, these approaches simply treat RGB difference as another video modality and train a different network to fuse with the RGB network. Instead, we aim to present a unified framework to capture appearance and motion information jointly, by generalizing the idea of temporal difference into a principled and efficient temporal module for end-to-end network

design.

In addition, we argue that both short-term and long-term temporal information are crucial for action recognition, in the sense that they are able to capture the distinctive and complementary properties of an action instance. Therefore, in our proposed temporal modeling mechanism, we present a unique two-level temporal modeling framework based on a holistic and sparse sampling strategy [38], termed as Temporal Difference Network (TDN). Specifically, in TDN, we consider two efficient forms of TDMs for motion modeling at different scales. For local motion modeling, we present a light-weight and low-resolution difference module to supply a single RGB with motion patterns via lateral connections, while for long-range motion modeling, we propose a multi-scale and bidirectional difference module to capture cross-segment variations for motion excitation. These two TDMs are systematically studied as modular building blocks for short-term and long-rang temporal structure extraction.

Our TDN provides a simple and general video-level motion modeling framework and could be instantiated with existing CNNs at a small extra computational cost. To demonstrate the effectiveness of TDN, we implement it with ResNets and perform experiments on two datasets: Kinetics and Something-Something. The evaluation results show that our TDN is able to yield a new state-of-the-art performance on both motion relevant Something-Something dataset and scene relevant Kinetics dataset, under the setting of using similar backbones. As shown in Figure 1, our best result is significantly better than previous methods on the dataset of Something-Something V1. We also perform detailed ablation studies to demonstrate the importance of temporal difference operation and investigate the effect of a specific design of TDM. In summary, our main contribution lies in the following three aspects:

- We generalize the idea of RGB difference to devise an efficient temporal difference module (TDM) for motion modeling in videos and provide an alternative to 3D convolutions by systematically presenting effective and detailed module design.
- Our TDN presents a video-level motion modeling framework with the proposed temporal difference module, with a focus on capturing both short-term and long-term temporal structure for video recognition.
- Our TDN obtains the new state-of-the-art performance on the datasets of Kinetics and Something-Something under the setting of using the same backbones. We also perform in-depth ablation study on TDM to provide some insights on our temporal difference modeling.

2. Related work

Short-term temporal modeling. Action recognition has attracted lots of research attention in the past few years. These methods could be categorized into two types: (1) two-stream CNNs [29] or its variants [7]: it used two inputs of RGB and optical flow to separately model appearance and motion information in videos with a late fusion; (2) 3D-CNNs [31, 12]: it proposed 3D convolution and pooling to directly learn spatiotemporal features from videos. Several variants tried to reduce the computation cost of 3D convolution by decomposing it into a 2D convolution and a 1D temporal convolution, for example R(2+1)D [33], S3D [41], P3D [25], and CT-Net [16]. Following this research line, several works focused on designing more powerful temporal modules and inserted them into a 2D CNN for efficient action recognition, such as TSM [19], TIN [28], TEINet [20], TANet [21], and TEA [18]. In addition, some methods tried to leverage the idea of two stream network to design a multi-branch architecture to capture both appearance and motion or context information, with a carefully designed temporal module or two RGB inputs sampled at different FPS, including Non-local Net [39], ARTNet [36], STM [13], SlowFast [6], and CorrelationNet [35]. Some recent works [5] tried network architecture search for video recognition. These works were clip-based architecture with a focus on short-term motion modeling by learning from a small portion of the whole video (e.g., 64 frames).

Long-term temporal modeling. Short-term clip-based networks fail to capture the long-range temporal structure. Several methods were proposed to overcome this limitation by stacking more frames with RNN [24, 3] or long temporal convolution [34], or using a sparse sampling and aggregation strategy [38, 44, 42, 9, 18]. Among these methods, temporal segment network (TSN) [38] turned out to be an effective long-range modeling framework and obtained the state-of-the-art performance with 2D CNNs on several benchmarks. However, TSN with 2D CNNs only performed temporal fusion at the last stage and failed to capture the finer temporal structure. StNet [9] proposed a local and global module to model temporal information hierarchically. V4D [42] extended the TSN framework by proposing a principled 4D convolutional operator to aggregate long-range information from different stages.

Temporal difference representation. Temporal difference operations appeared in several previous works for motion extraction, such as RGB Difference [38, 43, 23] and Feature Difference [20, 13, 18]. RGB difference turned out to be an efficient alternative modality to optical flow as motion representation [38, 43, 23]. However, they only treated RGB differently with another video modality and trained a separate network to fuse with RGB stream. The work of TEINet [20], TEA [18], and STM [13] employed a difference operation for network design. However, these two methods simply used a simple difference operator for single-level motion extraction and received less research attention than 3D convolutions.

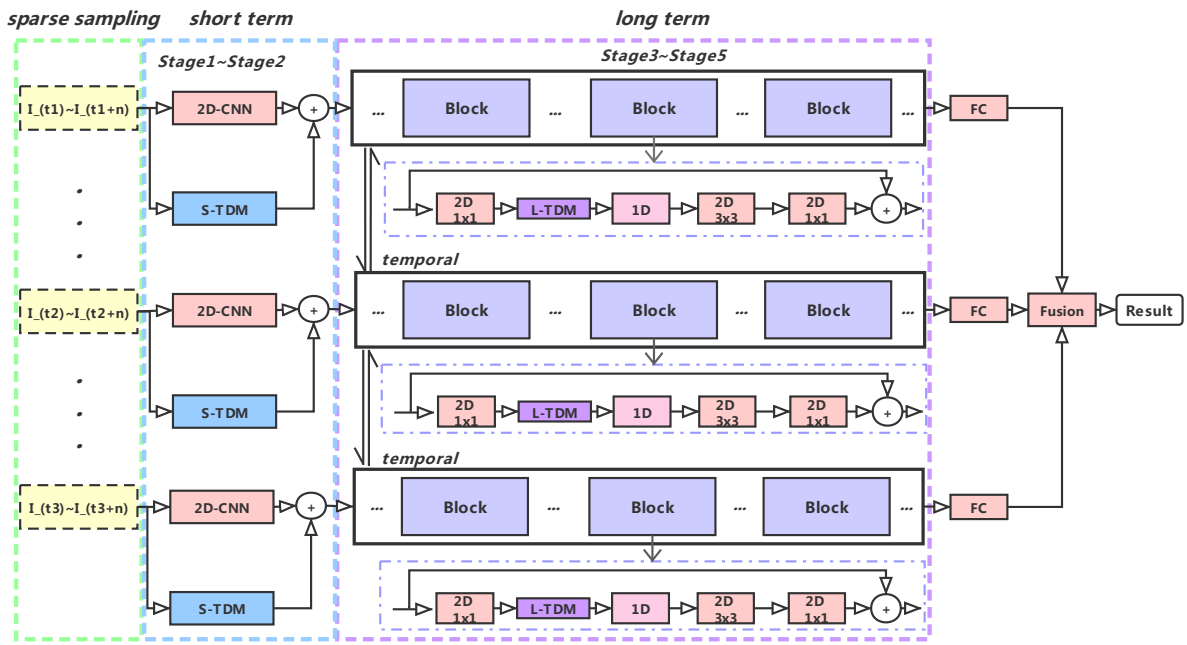

Figure 2. **Temporal Difference Network.** We present a video-level framework for learning action models from the entire video, coined as TDN. Based on the sparse sampling from multiple segments, our TDN aims to model both short-term and long-term motion information in our framework. The key contribution is to design an efficient short-term temporal difference module (S-TDM) and a long-term temporal difference module (L-TDM), to supply a 2D CNN with local motion information and enable long-range modeling across segments, respectively. CNNs share the same parameters on all segments. Details on both modules could be found in Figure 3.

Different from the existing methods, our proposed temporal difference network (TDN) is a video-level architecture of capturing both short-term and long-term information for end-to-end action recognition. Our key contribution is to introduce a temporal difference module (TDM) to explicitly compute motion information, and efficiently leverage it into our two-level motion modeling paradigm. We hope to improve and popularize these temporal difference-based modeling alternatives, which turns out to generally outperform 3D convolutions on two benchmarks with smaller FLOPs.

3. Temporal Difference Networks

In this section, we describe our Temporal Difference Network (TDN) in detail. First, we give an overview of the TDN framework, which is composed of a short-term and long-term temporal difference module (TDM). Then, we give a technical description of both modules. Finally, we provide the implementation detail to instantiate TDN with a ResNet backbone.

3.1. Overview

As shown in Figure 2, our proposed temporal difference network (TDN) is a video-level framework for learning action models by using the entire video information. Due to the limit of GPU memory, following TSN framework [38], we present a sparse and holistic sampling strategy for each video. Our key contribution is to leverage the temporal difference operator into the network design to explicitly capture both short-term and long-term motion information. Efficiency is our core consideration in temporal difference module (TDM) design, and we investigate two specific forms to accomplish the tasks of motion supplement in a local window and motion enhancement across different segments, respectively. These two modules are incorporated into the main network via a residual connection.

Specifically, each video V is divided into T segments of equal duration without overlapping. We randomly sample a frame from each segment and totally obtain T frames $\mathbf{I} = [I_1, \cdots, I_T]$, where the shape of $\mathbf{I}$ is $[T, C, H, W]$. These frames are separate fed into a 2D CNN to extract frame-wise features $\mathbf{F} = [F_1, \cdots, F_T]$, where $\mathbf{F}$ denotes the feature representation in the hidden layer and its dimension is $[T, C', H', W']$. The short-term TDM aims to supply these frame-wise representation $\mathbf{F}$ of early layers with local motion information to improve its representation power:

$$\textbf{Short term TDM}: \quad \hat{F}_i = F_i + \mathcal{H}(I_i), \tag{1}$$

where $\hat{F}_i$ denotes the enhanced representation by TDM, $\mathcal{H}$ denotes our short-term TDM, and it extracts local motion from adjacent frames around I_i. The long-term TDM aims at leveraging cross-segment temporal structure to enhance frame-level feature representation:

$$\textbf{Long term TDM}: \quad \hat{F}_i = F_i + F_i \odot \mathcal{G}(F_i, F_{i+1}), \tag{2}$$

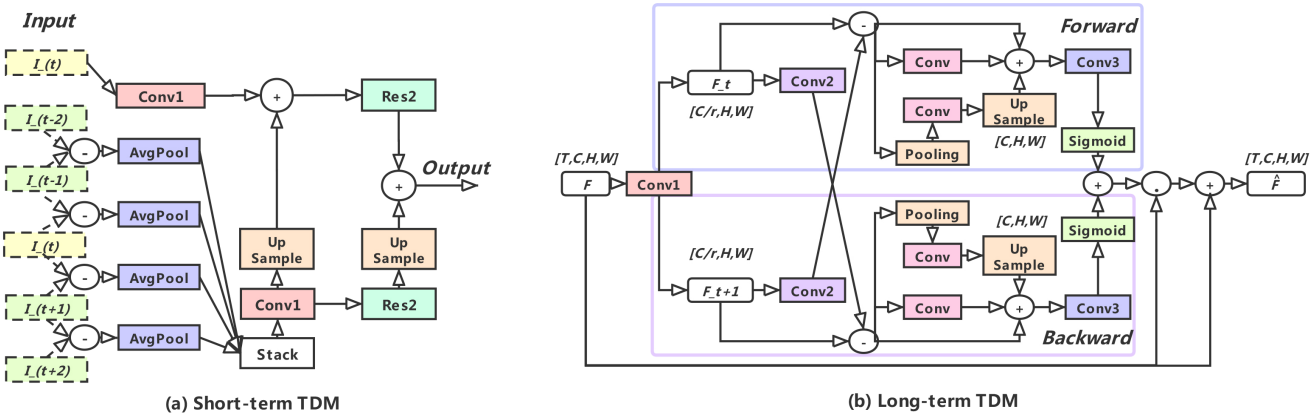

Figure 3. An illustration of the short-term TDM and long-term TDM. *Left*: our S-TDM operates on the stacked RGB difference and is fused with single RGB CNN via residual connection to capture the short-term motion. *Right*: our L-TDM presents a bi-directional and multi-scale attention mechanism to leverage cross-segment information to enhance the frame-wise representations. More details could be found in the text.

where $\mathcal{G}$ represents our long-term TDM, and in the current implementation, we only consider adjacent segment-level information for long-range temporal modeling in each long-term TDM. By stacking multiple long-term TDMs, we are able to capture temporal structure over a long scale. Details will be described in the next subsections.

3.2. Short-term TDM

We argue that adjacent frames are very similar in a local temporal window and directly stacking multiple frames for subsequent processing is inefficient. On the other hand, sampling a single frame from each window is able to extract appearance information, but fails to capture local motion information. Therefore, our short-term TDM chooses to supply a single RGB frame with a temporal difference to yield an efficient video representation, explicitly encoding both appearance and motion information.

Specifically, our short-term TDM operates at early layers of networks for low-level feature extraction and enables a single frame RGB to be aware of local motion via fusing temporal difference information. As shown in Figure 3, for each sampled frame I_i, we extract several temporal RGB difference in a local window centered at I_i, and then stack them along channel dimension $\mathbf{D}(I_i) = [D_{-2}, D_{-1}, D_1, D_2]$. Based on this representation, we present an efficient form of TDM:

$$\mathcal{H}(I_i) = \text{Upsample}(\text{CNN}(\text{Downsample}(\mathbf{D}(I_i)))), \quad (3)$$

where D represents the RGB difference around I_i, and CNN is the specific network for different stages. To keep the efficiency, we design a light-weight CNN module to operate on the stacked RGB difference $\mathbf{D}(I_i)$. It generally follows a low-resolution processing strategy: (1) downsample RGB difference by half with an average pooling, (2) extract motion features with a 2D CNN, (3) upsample motion features to match RGB features. This design comes from our observation that RGB difference exhibits very small values for most areas and only contains high response in motion salient regions. So, it is enough to use low-resolution architecture for this sparse signal without much loss of accuracy.

The information of short-term TDM is fused with the single RGB frame, so that the original frame-level representation is aware of motion pattern and able to better describe a local temporal window. We implement this fusion with lateral connections. We attach a fusion connection from short-term TDM to frame-level representation for each early stage (i.e., Stage 1-2 in our experiments). In practice, we also compare the residual connection with other fusion strategies as shown in the ablation study.

3.3. Long-term TDM

The frame-wise representation equipped with short-term TDM is powerful for capturing spatiotemporal information within a local segment (window). However, this representation is limited in terms of the temporal receptive field and thus fails to explore the long-range temporal structure for learning action models. Thus, our long-term TDM tries to use cross-segment information to enhance the original representation via a novel bidirectional and multi-scale temporal difference module.

In additional to efficiency, the missing-alignment of spatial location between long-range frames is another issue. Consequently, we devise a multi-scale architecture to smooth difference in large a receptive field before difference calculation. As shown in Figure 3, we first compress the feature dimension by a ratio r with a convolution for efficiency, and calculate the aligned temporal difference through adjacent segments:

$$C(F_i, F_{i+1}) = F_i - \text{Conv}(F_{i+1}), \quad (4)$$

where $C(F_i, F_{i+1})$ represents the aligned temporal difference for segment F_i, Conv is the channel-wise convolution for spatially smoothing and thus relieving the missing-alignment issue. Then, the aligned temporal difference undergoes through a multi-scale module for long-range motion information extraction:

$$M(F_i, F_{i+1}) = \text{Sigmd}(\text{Conv}(\sum_{j=1}^{N} \text{CNN}_j(C(F_i, F_{i+1})))), \quad (5)$$

where CNN_j at different spatial scales aims at extracting motion information from different receptive field, and $N = 3$ in practice. Their fusion could be more robust for the missing-alignment issue. In implementation, it involves three branches: (1) short connection, (2) a 3×3 convolution, and (3) a average pooling, a 3×3 convolution, and a bilinear upsampling. Finally, we utilize bidirectional cross-segment temporal difference to enhance fame level features as follows:

$$F_i \odot \mathcal{G}(F_i, F_{i+1}) = F_i \odot \frac{1}{2}[M(F_i, F_{i+1}) + M(F_{i+1}, F_i)], \quad (6)$$

where $\odot$ is the element-wise multiplication. We also combine the original frame level representation and enhance representation via a residual connection as in Eq. (2). Slightly different from short-term TDM, we employ the difference representation as an attention map to enhance frame level features, which is partially based on the observation that attention modeling is more effective for latter stage of CNNs. We also compare this implementation with other forms in the ablation study.

3.4. Exemplar: TDN-ResNet

As discussed above, our TDN framework is based on sparse sampling of TSN [38], which operates on a sequence of frames uniformly distributed over the entire video. Our TDN presents a two-level motion modeling mechanism, with a focus on capturing temporal information in a local-to-global fashion. In particular, as shown in Figure 2, we insert short-term TDMs (S-TDM) in early stages for finer and low-level motion extraction, and long-term TDMs (L-TDM) into latter stages for coarser and high-level temporal structure modeling.

We instantiate our TDN with a ResNet backbone [10]. Following the practice in V4D [42], the first two stages of ResNet are for short-term temporal information extraction within each segment by using S-TDMs, and the latter three stages of ResNet are equipped with L-TDMs for capturing long-range temporal structure across segments. For local motion modeling, we add both residual connections between S-TDM and the main network for Stage 1 and Stage 2. For long term motion modeling, we add L-TDM and a temporal convolution in each residual block of Stages 3-5. In practice, the final TDN-ResNet only increases the FLOPs over the original 2D TSN-ResNet by around 9%.

4. Experiments

In this section, we present the experiment results of our TDN framework. First, we describe the evaluation datasets and implementation details. Then, we perform ablation studies on the design of our TDN. Next, we compare our TDN with the state-of-the-art methods. Finally, we show some visualization results to further analyze our TDN.

4.1. Datasets and implementation details

Video datasets. We evaluate our TDN on two video datasets, which pay attention to different aspects of an action instance for recognition. **Kinetics-400** [15] is a large-scale YouTube video dataset and has around 300k trimmed videos covering 400 categories. The Kinetics dataset contains activities in daily life and some categories are highly correlated with interacting objects or scene context. We train our TDN on the training data (around 240k videos) and report performance on the validation data (around 20k videos). **Something-Something** [8] is a large-scale dataset created by crowdsourcing. The videos are collected by performing the same action with different objects so that action recognition is expected to focus on the motion property instead of objects or scene context. The first version contains around 100k videos over 174 categories, while the second version is with more videos, containing around 169k videos in training set and 25k videos in validation set. We report performance on the validation set of Something-Something V1 & V2.

Training and testing. In experiments, we use ResNet50 and ResNet101 to implement our TDN framework, and sample $T = 8$ or $T = 16$ frames from each video. Following common practice [6, 39], during training, each video frame is resized to have shorter side in $[256, 320]$ and a crop of 224×224 is randomly cropped. We pre-train our TDN on the ImageNet dataset [2]. The batch size is 128 and the initial learning rate is 0.02. The total training epoch is set as 100 in the Kinetics dataset and 60 in the Something-Something dataset. The learning rate will be divided by a factor of 10 when the performance on validation set saturates. For testing, the shorter side of each video is resized to 256. We implement two kinds of testing scheme: **1-clip and center-crop** where only a center crop of 224×224 from a single clip is used for evaluation, and **10-clip and 3-crop** where three crops of 256×256 and 10 clips are used for testing. The first testing scheme is with high efficiency while the second one is for improving accuracy with a denser prediction scheme.

4.2. Ablation studies

We perform ablation studies on the Something-Something V1 dataset. For these evaluations, we use the testing scheme of 1 clip and center crop, and report the Top1

S-TDM	L-TDM	FLOPs	Top1
concat	avg	36.2G	41.5%
concat	diff✓	36.2G	51.4%
diff✓	avg	35.9G	51.6%
diff✓	diff✓	**35.9G**	**52.3%**

(a) **Study on the effect of difference operator**: We compare with baselines by directly stacking or averaging temporal frames, which are worse than temporal difference operator for both short-term and long-term modeling.

Fusion	Top1
$F \odot \mathcal{H}$	43.7%
$F + F \odot \mathcal{H}$	47.6%
$F + \mathcal{H}$	**52.3%**
$F \odot \mathcal{H}_{channel}$	47.3%
$F + F \odot \mathcal{H}_{channel}$	47.9%

(b) **Study on S-TDM**: We compare different implementation forms of S-TDM, including spatiotemporal attention, channel attention, combination of residual connection and attention.

Fusion	Multi-scale	bidirect.	Top1
$F + \mathcal{G}$	✓	✓	44.1%
$F + F \odot \mathcal{G}_{channel}$		✓	50.9%
$F + F \odot \mathcal{G}$	✓		50.0%
$F + F \odot \mathcal{G}$		✓	49.7%
$F + F \odot \mathcal{G}$	✓	✓	**52.3%**

(c) **Study on L-TDM**: We compare different implementation forms of L-TDM, with a focus on multi-scale representation and bidirectional difference, and different fusion strategies, including residual connection, channel attention, and spatiotemporal attention.

S-TDM	L-TDM	FLOPs	Top1
-	-	33G	46.6%
Stage 1	Stage 2-5	35G	50.6%
Stage 1-2	Stage 3-5	36G	**52.3%**
Stage 1-3	Stage 4-5	38G	51.7%

(d) **Study on the location of S-TDM and L-TDM**: We place S-TDMs and L-TDMs at different stages of ResNet50. The results imply that it obtains the best performance when stages 1-2 focus on short-term modeling and stages 3-5 focus on long-term modeling.

S-TDM		L-TDM			Top1
Conv1	Res2	Res3	Res4	Res5	
					46.6%
✓					51.3%
✓	✓				51.5%
		✓	✓	✓	49.9%
✓	✓	✓	✓	✓	**52.3%**

(e) **S-TDM vs. L-TDM**: We compare the effectiveness of S-TDM and L-TDM. The results imply that S-TDM is slightly better than L-TDM when simply using a single types of TDM, and S-TDM and L-TDM are complementary to each other.

Model	FLOPs	Top1	Top5
T-Conv [33]	33G	47.5%	77.5%
T-Conv++ [33]	165G	48.2%	79.1%
TSM [19]	33G	47.1%	76.2%
TSM++ [19]	165G	47.6%	77.9%
TEINet [20]	33G	48.4%	77.2%
TEINet++ [20]	165G	49.0%	79.0%
TDM	36G	**52.3%**	**80.6%**

(f) **Comparison with other temporal modules**: We compare TDM with several temporal modules: Temporal convolution, TSM, and TEINet. For fair comparison, we report the result with 8 frame and 40 for each temporal module (++ for 40 frames). Our TDM is better than previous temporal modules.

Table 1. Ablations on **Something-Something V1** with 8-frame TDN-ResNet50. We show top-1 classification accuracy (%), and computational cost measured in FLOPs (floating-point operations, in # of multiply-adds) for a 1-clip and center-crop input of size 224×224.

accuracy. We also compare with other temporal modeling modules to demonstrate the effectiveness of TDM.

Study on the effect of difference operation. We begin our oblation study by exploring the effectiveness of temporal difference operation in our TDM. We implement fairly comparable baselines by simply removing temporal difference operation in S-TDM and replacing temporal difference with taking average in L-TDM. Table 1a shows the results of various settings with temporal difference or without temporal difference. It can be seen that simply stacking and taking average to fuse temporal information will greatly decrease the recognition accuracy by around 10%. We analyze that these temporal fusion strategies without difference operation would make the network to over-fit static information and fail to capture temporal variation in videos. Adding temporal difference in both S-TDM and L-TDM contributes to better accuracy and their combination obtains the best performance.

Study on short-term TDM. We compare different forms of short-term TDM (S-TDM). We add long-term TDM (L-TDM) for all latter stages and place variations of S-TDM in early stages. As shown in Table 1b, we first compare different fusion strategies to combine difference representation with RGB features in S-TDM: (1) attention with element-wise multiplication, (2) addition with attention, (3) only addition. We can see that our S-TDM with simply addition yields the best performance and the other attention-based fusion might destroy the pre-trained feature correspondence. In addition, we try to use RGB difference representation to learn the channel attention weight just as TEINet [21], and its performance is also worse than our proposed S-TDM (47.3% vs. 52.3%). In the remaining study, we use the addition form of S-TDM by default.

Study on long-term TDM. We employ short-term TDM for the early stages, and compare with different forms of long-term TDM (L-TDM) placed on the latter stages. The results are reported in Table 1c. For L-TDM design, we first compare with two baseline architecture: (1) no attention modeling in Eq. (2) and directly adding the difference representation into frame-level features; (2) channel attention modeling just as TEA [18]. It is observed that our proposed spatiotemporal attention form of L-TDM is better than no attention (52.3% vs.44.1%) and channel attention (52.3% vs. 50.9%). Then, we investigate the effectiveness of multi-scale architecture in difference feature extraction and it is able to improve performance from 49.7% to 52.3%, which confirms its effectiveness of large receptive field for difference feature extraction. Finally, we compare the performance of bidirectional difference with one-directional difference, and it helps to improve performance by 2.3%.

Study on the location of S-TDM and L-TDM . We perform the ablation study on which stage to use short-term TDM (S-TDM) or long-term TDM (L-TDM). The results are shown in Table 1d. From these results, we see that

Method	Backbone	Frames	GFLOPs	Sth-Sth V1 Top1	Sth-Sth V1 Top5
TSN-RGB [38]	BNInception	8	16	19.5	-
TRN-Multiscale [44]	BNInception	8	33	34.4	-
S3D-G [41]	Inception	64	71.38	48.2	78.7
TSM [19]	ResNet50	8+16	98	49.7	78.5
TEINet [20]	ResNet50	8+16	99	52.5	-
TANet [21]	ResNet50	8+16	99	50.6	79.3
TEA [18]	ResNet50	16	70	51.9	80.3
TAM [4]	bLResNet50	16×2	47.7	48.4	78.8
ECO$_{En}$Lite [46]	BNIncep+R18	92	267	46.4	-
I3D [1]	ResNet50	32×2	306	41.6	72.2
NL I3D+GCN [40]	R50+GCN	32×2	606	46.1	76.8
GST [22]	ResNet50	16	59	48.6	77.9
STM [13]	ResNet50	16×30	67×30	50.7	80.4
V4D [42]	ResNet50	8×4	167.6	50.4	-
SmallBigNet [17]	ResNet50	8+16	157	50.4	80.5
CorrNet [35]	ResNet50	32×10	115×10	49.3	-
TDN (Ours)	ResNet50	8	36	52.3	80.6
TDN (Ours)	ResNet50	16	72	53.9	82.1
TDN (Ours)	ResNet50	8+16	108	**55.1**	**82.9**
CorrNet [35]	ResNet101	32×30	224×30	51.7	-
CorrNet [35] [1]	ResNet101	32×30	224×30	53.3	-
GSM [30]	Inception V3	fusion	268	55.2	-
TDN (Ours)	ResNet101	8	66	54.1	81.9
TDN (Ours)	ResNet101	16	132	55.3	83.3
TDN (Ours)	ResNet101	8+16	198	**56.8**	**84.1**
Method	**Backbone**	**Frames**	**GFLOPs**	**Sth-Sth V2 Top1**	**Sth-Sth V2 Top5**
TRN-Multiscale [44]	BNInception	8	33	48.8	77.6
TAM [4]	bLResNet50	16×2	47.7	61.7	88.1
TSM [19]	ResNet50	16×6	65×6	63.4	88.5
GST [22]	ResNet50	16	59	62.6	87.9
STM [13]	ResNet50	16×30	67×30	64.2	89.8
SmallBigNet [17]	ResNet50	8+16	157	63.3	88.8
TEINet [19]	ResNet50	8+16	98	65.5	89.8
TANet [21]	ResNet50	24×6	99 ×6	66.0	90.1
TDN (Ours)	ResNet50	8	36	64.0	88.8
TDN (Ours)	ResNet50	16	72	65.3	89.5
TDN (Ours)	ResNet50	8+16	108	**67.0**	**90.3**
TDN (Ours)	ResNet101	8	66	65.8	90.2
TDN (Ours)	ResNet101	16	132	66.9	90.9
TDN (Ours)	ResNet101	8+16	198	**68.2**	**91.6**

Table 2. **Comparison with the state-of-the-art methods on Something-Something V1 and V2**. We instantiate our TDN with the backbones of ResNet50 and ResNet101 for evaluation. We compare with other methods with similar backbones under the *1-clip and center crop setting*. "-" indicates the numbers are not available for us. [1] Pre-trained on Sports1M.

adding more S-TDMs into the main network will increase the network computational cost slightly due to its feature extraction for temporal difference representation. The setting of using S-TDM in stages 1-2 and L-TDM in stages 3-5 obtains the best recognition accuracy and the computational cost is also reasonable.

Short-term vs. long-term modeling. We conduct comparative study to separately investigate the effectiveness of S-TDM and L-TDM. The results are summarized in Table 1e. We first report the performance of baseline without

Method	Backbone	Frames	GFLOPs	Top1	Top5
TSN [38]	InceptionV3	25×1×10	3.2×250	72.5	90.2
S3D-G [41]	InceptionV1	64×10×3	71.4×30	74.7	93.4
R(2+1)D [33]	ResNet34	32×10×1	152×10	74.3	91.4
TSM [19]	ResNet50	16×10×3	65×30	74.7	91.4
TEINet [20]	ResNet50	16×10×3	66×30	76.2	92.5
TEA [18]	ResNet50	16×10×3	70×30	76.1	92.5
TAM [4]	bLResNet50	48×3×3	93.4×9	73.5	91.2
TANet [21]	ResNet50	16×4×3	86×12	76.9	92.9
ARTNet [36]	ResNet18	16 ×25×10	23.5×250	70.7	89.3
I3D [1]	InceptionV1	64×N/A×N/A	108×N/A	72.1	90.3
NL I3D [39]	ResNet50	128×10×3	282×30	76.5	92.6
SlowOnly [6]	ResNet50	8×10×3	41.9×30	74.8	91.6
SlowFast [6]	ResNet50	(4+32)×10×3	36.1×30	75.6	92.1
SlowFast [6]	ResNet50	(8+32)×10×3	65.7×30	77.0	92.6
SmallBigNet [17]	ResNet50	8×10×3	57×30	76.3	92.5
CorrNet [35]	ResNet50	32×10×1	115×10	77.2	-
TDN (Ours)	ResNet50	8×10×3	36×30	76.6	92.8
TDN (Ours)	ResNet50	16×10×3	72×30	77.5	93.2
TDN (Ours)	ResNet50	(8+16)×10×3	108×30	**78.4**	**93.6**
NL I3D [39]	ResNet101	128×10×3	359×30	77.7	93.3
ip-CSN [32]	ResNet101	32×10×3	83.0×30	76.7	92.3
SlowFast [6]	ResNet101	(8+32)×10×3	106×30	77.9	93.2
SlowFast [6]	ResNet101	(16+64)×10×3	213×30	78.9	93.5
SmallBigNet [17]	ResNet101	32×4×3	418×12	77.4	93.3
CorrNet [35]	ResNet101	32×10×3	224×30	79.2	-
TDN (Ours)	ResNet101	8×10×3	66×30	77.5	93.6
TDN (Ours)	ResNet101	16×10×3	132×30	78.5	93.9
TDN (Ours)	ResNet101	(8+16)×10×3	198×30	**79.4**	**94.4**
SlowFast [6]	R101+NL	(16+64)×10×3	234×30	**79.8**	93.9
X3D [5]	X3D-XL	16×10×3	48.4×30	79.1	93.9

Table 3. **Comparison with the state-of-the-art methods on the validation set of Kinetics-400**. We instantiate our TDN with the backbones of ResNet50 and ResNet101. For fair comparison, we compare with the other methods by using the similar backbones without pre-training on extra videos. "-" indicates the numbers are not available for us.

S-TDM or L-TDM, namely only with 1D temporal convolutions in latter stages for temporal modeling, and its accuracy is 46.6%. Then we separately add S-TDM and L-TDM into the baseline, and they obtain the performance of 51.5% and 49.9%. The superior performance of S-TDM to L-TDM might be ascribed to the fact that local motion information is crucial for action recognition. Finally, combining S-TDM and L-TDM could boost performance to 52.3%, which implies the complementarity of two modules.

Comparison with other temporal modules. Finally, we compare our proposed TDM with other temporal modeling methods, and the results are reported in Table 1f. We compare our TDM with three temporal modules: temporal convolution [33], TSM [19], and TEINet [20]. First, these methods all use the ResNet50 as backbones and 8 frames as input. In this setting, their FLOPs are similar to our TDN. We find that the performance of our TDN is much better than those baselines with similar FLOPs, demonstrating the effectiveness of explicit temporal difference operation. Then, for fair comparison, we also implement other tem-

poral modules taking the same number of frames as ours (i.e., 40 frames denoted by ++), and we observe that simply inputting more frames will not contribute much to improve recognition accuracy. We analyze that these temporal modules still lack sufficient modeling capacity to well capture fine-grained motion information and thus more frames will make them over-fit with appearance more seriously. On the other hand, thanks to temporal difference operation, our TDM is able to focus more on the motion information and thus improve the recognition accuracy with more frames.

4.3. Comparison with the state of the art

After the ablation study of 8-frame TDN on Something-Somthing V1 dataset, we directly transfer its optimal setting to the datasets of Something-Something V2 and Kinetics-400. In this subsection, we compare our TDN with those state-of-the-art methods on these benchmarks. As expected, sampling more frames can further improve the accuracy, but also increases the FLOPs. We report the performance of both 8-frame TDN and 16-frame TDN. For fair comparison, we simply list the performance of methods solely using RGB without pre-training on extra video datasets.

The results are summarized in Table 2 and Table 3. For fair comparison with previous methods, we use 1 clip and center crop testing scheme on the Something-Something dataset and 10 clips and 3 crops for testing on the Kinetics-400 dataset. We first compare with 2D CNN based baselines with late fusion for long-range temporal modeling such as TSN [38] and TRN [44], and see that our TDN outperforms these baseline methods significantly on both datasets. Then, we compare our TDN with 2D CNN with temporal modules for all stages, such as S3D [41], R(2+1)D [33], TSM [19], TEINet [20], TANet [21], TAM [4], and GSM [30], and our TDN consistently outperforms them on both datasets, demonstrating the effectiveness of TDM in temporal modeling for action recognition. After this, we compare with more recent 3D CNNs based methods, such as I3D [1], Non-local I3D [39], and SlowFast [6], and our TDN can still obtain slightly better performance than those methods, with a relatively smaller computational cost. Finally, we compare more recent video recognition networks, such as SmallBigNet [17], V4D [42], CorrelationNet [35], and X3D [5]. Our best result significantly outperforms previous methods on Something-Something V1 and is on par with the previous best performance on the Kinetics dataset. The best performance on the Kinetics dataset is the combination of SlowFast and Non-local Net, which is slightly better than ours for Top1 accuracy yet with lower Top5 accuracy and higher FLOPs.

4.4. Visualization of activation maps

We visualize the class activation maps with Grad-CAM [45, 27] and results are shown in Figure 4. In this visualization, we take 8 frames as input and only plot the activation maps in the center frames. These visualization results indicate that baseline with only temporal convolutions fails to focus on motion-salient regions, while our TDN is able to localize more action-relevant regions, thanks to our proposed TDMs for short-term and long-term temporal modeling. For example, our TDN pays more attention to the hand motion with interaction objects, while the temporal convolution may only focus on the background. More visualization examples and analysis could be found in the supplementary material.

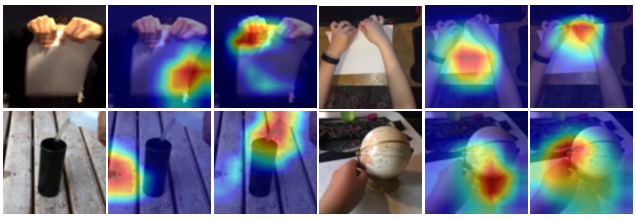

Figure 4. Visualization of activation maps with CAM. *Left*: video, *Middle*: baseline, *Right*: TDN. In this visualization, we train a 8-frame network with TDMs (TDN) or temporal convolutions (Baseline). For simplicity, we only visualize the CAM on the center frames. More visualization examples on 8 frames could be found in supplementary material.

5. Conclusion

In this paper, we have presented a new video-level framework, termed as TDN, for learning action models from the entire video. The core contribution of TDN is to generalize temporal difference operator into efficient and general temporal modules (TDM) with specific designs, for capturing both short-term and long-term temporal information in a video. We present two customized forms for the implementation of TDMs and systematically assess their effects on temporal modeling. As demonstrated on the Kinetics-400 and Something-Something dataset, our TDN is able to yield superior performance to previous state-of-the-art methods of using similar backbones.

In addition, we present an in-depth ablation study on TDMs to investigate the effect of temporal difference operation, and demonstrate that it is more effective to extract fine-grained temporal information than a standard 3D convolution with more frames. We hope our analysis provides more insights about temporal difference operation, and TDM might provide an alternative to 3D convolution for temporal modeling in videos.

Acknowledgements. This work is supported by the National Science Foundation of China (No. 62076119, No. 61921006), Program for Innovative Talents and Entrepreneur in Jiangsu Province, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733, 2017. 7, 8

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[3] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015. 2

[4] Quanfu Fan, Chun-Fu (Richard) Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. In *NIPS*, pages 2261–2270, 2019. 7, 8

[5] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, pages 203–213, 2020. 2, 7, 8

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6201–6210, 2019. 1, 2, 5, 7, 8

[7] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016. 2

[8] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, pages 5843–5851, 2017. 1, 5

[9] Dongliang He, Zhichao Zhou, Chuang Gan, Fu Li, Xiao Liu, Yandong Li, Limin Wang, and Shilei Wen. Stnet: Local and global spatial-temporal modeling for action recognition. In *AAAI*, pages 8401–8408, 2019. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[11] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artif. Intell.*, 17(1-3):185–203, 1981. 1

[12] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. In *ICML*, pages 495–502, 2010. 1, 2

[13] Boyuan Jiang, Mengmeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. STM: spatiotemporal and motion encoding for action recognition. In *ICCV*, pages 2000–2009, 2019. 2, 7

[14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Fei-Fei Li. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 1

[15] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. 5

[16] Kunchang Li, Xianhang Li, Yali Wang, Jun Wang, and Yu Qiao. CT-net: Channel tensorization network for video classification. In *ICLR*, 2021. 2

[17] Xianhang Li, Yali Wang, Zhipeng Zhou, and Yu Qiao. Smallbignet: Integrating core and contextual views for video classification. In *CVPR*, pages 1092–1101, 2020. 7, 8

[18] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, pages 909–918, 2020. 2, 6, 7

[19] Ji Lin, Chuang Gan, and Song Han. TSM: temporal shift module for efficient video understanding. In *ICCV*, pages 7082–7092, 2019. 1, 2, 6, 7, 8

[20] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *AAAI*, 2020. 1, 2, 6, 7, 8

[21] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. TAM: temporal adaptive module for video recognition. *CoRR*, abs/2005.06803, 2020. 2, 6, 7, 8

[22] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *ICCV*, pages 5512–5521, 2019. 7

[23] Joe Yue-Hei Ng and Larry S Davis. Temporal difference networks for video action recognition. In *WACV*, pages 1587–1596, 2018. 2

[24] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015. 2

[25] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. *ICCV*, pages 5534–5542, 2017. 1, 2

[26] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. In *CVPR*, pages 12056–12065, 2019. 1

[27] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *IJCV*, 128(2):336–359, 2020. 8

[28] Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. *AAAI*, 34:11966–11973, 04 2020. 2

[29] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 1, 2

[30] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *CVPR*, pages 1099–1108, 2020. 7, 8

[31] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1, 2

[32] Du Tran, Heng Wang, Matt Feiszli, and Lorenzo Torresani. Video classification with channel-separated convolutional networks. In *ICCV*, pages 5551–5560, 2019. 7

[33] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 1, 2, 6, 7, 8

[34] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1510–1517, 2018. 2

[35] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *CVPR*, pages 352–361, 2020. 2, 7, 8

[36] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, pages 1430–1439, 2018. 2, 7

[37] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015. 1

[38] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016. 1, 2, 3, 5, 7, 8

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 2, 5, 7, 8

[40] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, pages 413–431, 2018. 1, 7

[41] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, volume 11219, pages 318–335. Springer, 2018. 1, 2, 7, 8

[42] Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R Scott, and Limin Wang. V4d: 4d convolutional neural networks for video-level representation learning. In *ICLR*, 2020. 2, 5, 7, 8

[43] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Recognize actions by disentangling components of dynamics. In *CVPR*, pages 6566–6575, 2018. 1, 2

[44] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, volume 11205, pages 831–846. Springer, 2018. 2, 7, 8

[45] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 8

[46] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: efficient convolutional network for online video understanding. In *ECCV*, pages 713–730, 2018. 1, 7