TEMPORAL DIFFERENCE NETWORKS FOR ACTION RECOGNITION

Anonymous authors
Paper under double-blind review

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 datasets of Something-Something V1 & V2 and Kinetics-400 under the setting of using similar backbones. In addition, we present some visualization results on our TDN and try to provide new insights on temporal difference operation.

1 INTRODUCTION

Deep neural networks have witnessed great progress for action recognition in videos (Karpathy et al., 2014; Simonyan & Zisserman, 2014; Wang et al., 2016; Tran et al., 2015; Feichtenhofer et al., 2019). 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 (Simonyan & Zisserman, 2014), 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 (Ji et al., 2010; Tran et al., 2015) or temporal convolutions (Tran et al., 2018; Xie et al., 2018; Qiu et al., 2017) to implicitly learn motion features from RGB frames. However, 3D convolutions often lack specific consideration in 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.

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 (Horn & Schunck, 1981), and has shown effectiveness in action recognition by using RGB difference as an approximate motion representation (Wang et al., 2016; Zhao et al., 2018). Following this research line, we focus on generalizing the idea of temporal difference into a principled temporal module for network design. In addition, we argue that both short-term and long-term temporal information are crucial for action recognition, in sense that they are able to capture 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, 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 kinds of TDMs are systematically studied as a principled building block 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 ResNet50 and perform experiments on two datasets: Kinetics and Something-Something. The evaluation results show that our TDN is able to yeild 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. 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 principled 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 a new state-of-the-art performance on the datasets of Kinetics and Something-Something under the setting of using similar backbones. We also give several visualization results to analyze 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 (Simonyan & Zisserman, 2014) or its variants (Feichtenhofer et al., 2016): 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 (Tran et al., 2015; Ji et al., 2010): 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 (Tran et al., 2018), S3D (Xie et al., 2018), and P3D (Qiu et al., 2017). Following this research line, several works focused on designing a more powerful temporal module and inserted it into a 2D CNN for efficient action recognition, such as Non-local net (Wang et al., 2018b), TSM (Lin et al., 2019), TIN (Shao et al., 2020) and TEINet (Liu et al., 2020). 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 information, including ARTNet (Wang et al., 2018a), STM (Jiang et al., 2019) and SlowFast (Feichtenhofer et al., 2019). 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 fails to capture long-range temporal structure for video recognition. Several methods were proposed to overcome this limitation by stacking more frames with RNN (Ng et al., 2015; Donahue et al., 2015) or long temporal convolution (Varol et al., 2018), or using a sparse sampling and aggregation strategy (Wang et al., 2016; Zhou et al., 2018; Zhang et al., 2019; He et al., 2019). Among these methods, temporal segment network (TSN) (Wang et al., 2016) 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 last stage and failed to capture finer temporal structure. StNet (He et al., 2019) proposed a local and global module to model temporal information hierarchically. V4D (Zhang et al., 2019) 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 (Wang et al., 2016; Zhao et al., 2018) and Feature Difference (Liu et al., 2020; Jiang et al., 2019). RGB difference turned out to be an efficient alternatives to optical flow motion representation in two-stream CNNs (Wang et al., 2016; Zhao et al., 2018). The work of TEINet (Liu et al., 2020) and STM (Jiang et al., 2019) employed a dif-

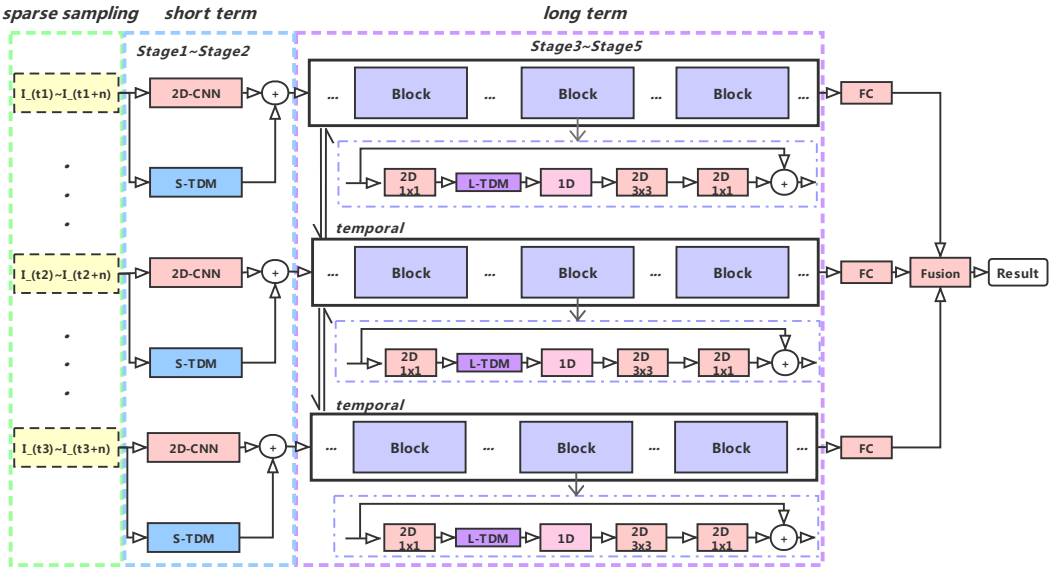

Figure 1: **Temporal Difference Network.** We present a video-level framework for learning action models from the entire video, coined as TDN. Based on the sparse sampling from multiple segments, our TDN aims to model both short-term and long-term motion information in our framework. The key contribution 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 2.

ference operation for network design. However, these two methods simply used a simple difference operator for short-term motion extraction and received less research attention than 3D convolutions.

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 this new temporal 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 details. First, we give an overview on the TDN framework, that is composed of a short-term and long-term temporal difference modules (TDM). Then, we give a technical description on both modules. Finally, we provide the implementation detail to instantiate TDN with a ResNet50 backbone.

3.1 Overview

As shown in Figure 1, 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 (Wang et al., 2016), we present a sparse and holistic sampling strategy for each video. Our key contribution is to leverage temporal difference operator into 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. These two modules are incorporated into the main network via a residual connection.

Specifically, each video is divided into T segments of equal duration without overlapping. We randomly sample a frame from each segment and totally obtain T frames $\mathbf{X} = [X_1, \cdots, X_T]$, where the shape of $\mathbf{X}$ is $[T, C, H, W]$. These frames are separate fed into a 2D CNN to extract frame-wise

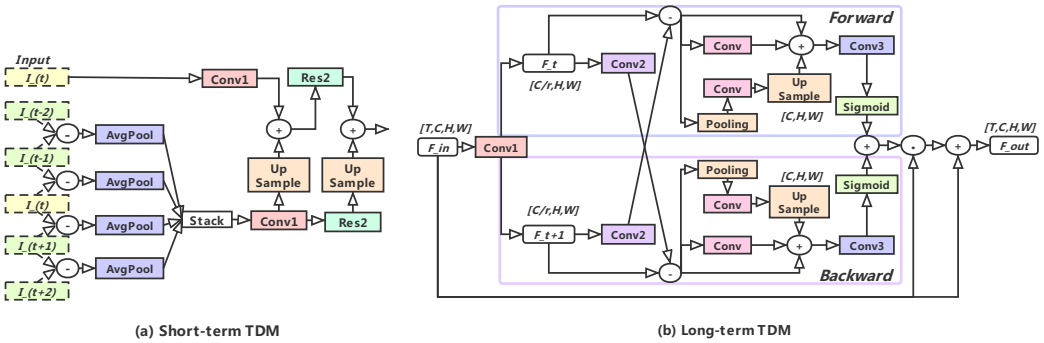

Figure 2: An illustration of the short-term TDM and long-term TDM.

features $\mathbf{F} = [F_1, \cdots, F_T]$, where $\mathbf{F}$ denotes the feature representation in a hidden layer and its dimension is $[T, C', H', W']$. The short-term TDM aims to supply these frame-wise representation of early layers with local motion information to improve its representation power:

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

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

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

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. Next, we will give the technical details of these two modules.

3.2 SHORT-TERM TDM

We argue that adjacent frames are very similar in a local temporal window and directly stacking multiple frames together for subsequent processing is inefficient. On the other hand, a single frame from each segment 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 local representation, encoding both appearance and motion information.

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

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

where D_i represents the RGB difference around X_i. To keep the efficiency, we design a light-weight CNN module to operate on the stacked RGB difference $\mathbf{D}(X_i)$. It general 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 form 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 a feasible to use low-resolution architecture to process 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 use the residual connection to implement this fusion and also compare with other fusion strategies as shown in 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. However, this representation is limited in terms of temporal receptive field and thus fails to explore 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. As shown in Figure 2, we first compress the feature dimension by a ratio r with a convolution for efficiency, and calculate the aligned temporal difference through adjacent segments:

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

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:

$$M(F_i, F_{i+1}) = \mathrm{Sigmoid}(\mathrm{Conv}(\sum_{j=1}^{N} \mathrm{CNN}_j(C(F_i, F_{i+1})))), \tag{5}$$

where CNN_j at different spatial scales aims at extracting motion information from different receptive filed, and $N = 3$ in practice. Their fusion could be more robust for 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:

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

where $\odot$ is the element-wise multiplication. We also combine segment level representation and long-term TDM via residual connection as in Eq. (2). Slightly different from short-term TDM, we employ the motion 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 ablation study.

3.4 EXEMPLAR: TDN-RESNET50

After introducing short-term and long-term TDMs, we are ready to describe to how incorporate them into the existing video architecture. As discussed above, our TDN framework is based on sparse sampling of TSN (Wang et al., 2016), 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, 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.

To keep a balance between efficiency and accuracy, we instantiate our TDN with a ResNet50 backbone (He et al., 2016). Following the practice in V4D (Zhang et al., 2019), the first two stages of ResNet50 are for short-term temporal information extraction within each segment by using S-TDMs, and the latter three stages of ResNet50 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 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 framework only increases the FLOPs over the original 2D 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 study on the design of our TDN. After that, we compare our TDN with the existing state-of-the-art methods. Finally, we show some visualization results to further analyze our TDN framework.

Fusion	Top1
$F \odot \mathcal{H}$	43.7%
$F + F \odot \mathcal{H}$	47.6%
$F + \mathcal{H}$	**51.3%**
$F \odot \mathcal{H}_{channel}$	47.3%
$F + F \odot \mathcal{H}_{channel}$	47.9%

(a) Study on S-TDM.

Fusion	Multi-scale	bidirectional	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%
$F + F \odot \mathcal{G}$	✓	✓	**51.3%**

(b) Study on L-TDM.

S-TDM		L-TDM			Top1
Conv1	Res2	Res3	Res4	Res5	
					45.2%
✓					49.3%
✓	✓				49.5%
		✓	✓	✓	48.9%
✓	✓	✓	✓	✓	**51.3%**

(c) S-TDM vs. L-TDM.

model	FLOPs	Top1	Top5
TSN (Wang et al., 2016)	33G	19.7%	46.5%
T-Conv (Tran et al., 2018)	33G	46.2%	75.1%
TSM (Lin et al., 2019)	33G	45.6%	74.2%
TEINet (Liu et al., 2020)	33G	47.4%	76.6%
TEA (Li et al., 2020)	35G	48.9%	78.1%
TAM (Fan et al., 2019)	-	46.1%	-
TDM	36G	**51.3%**	**79.3%**

(d) Comparison with temporal modules.

4.1 Datasets and implementation details

Video datasets. We evaluate our TDN on two video datasets, which focus on different aspects of an action instance for recognition. **Kinetics-400** (Kay et al., 2017) is a large-scale YouTube video dataset, and has around 300k trimmed videos covering 400 categories. The Kinetics dataset contains activities in our 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** (Goyal et al., 2017) is a large-scale dataset created by crowdsourcing. These videos are collected by performing the same action with different objects so that action recognition methods are expected to focus on the motion property instead of objects. The first version contains around 100k videos over 174 categories, while the second version are 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 to implement our TDN framework, and we try to sample $T = 8$ or $T = 16$ frames from each video. Following common practice (Feichtenhofer et al., 2019; Wang et al., 2018b), during training, each video is resized to have shorter side in $[256, 320]$ and a crop of 224×224 was randomly cropped. We pretrain our TDN on the ImageNet dataset (Deng et al., 2009). The batch size is 128 and initial learning rate is 0.02 for 8-frame TDN, while 64 and 0.01 for 16-frame TDN. The total training epoch is set as 100 in the Kinetics dataset and 50 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, each video is resized to have shorter size as 256. We try 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 prediction fusion.

4.2 Ablation study

We perform ablation study on TDN design in the Something-Something V1 dataset. For these evaluations, we use the testing scheme of 1 clip and center crop, and report the Top1 accuracy. We also compare with several temporal modeling modules to demonstrate the effectiveness of TDM.

Study on short-term TDM. We begin our experiments by comparing different forms of short-term TDM (S-TDM). In this study, we add long-term TDM (L-TDM) for all latter stages and place variations of S-TDM in early stages. As shown in Table 1a, 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 a channel attention weight just as SENet (Hu et al., 2018) and its performance is also worse than our proposed S-TDM (47.3% vs. 51.3%). In the remaining study, we use the addition form of S-TDM by default.

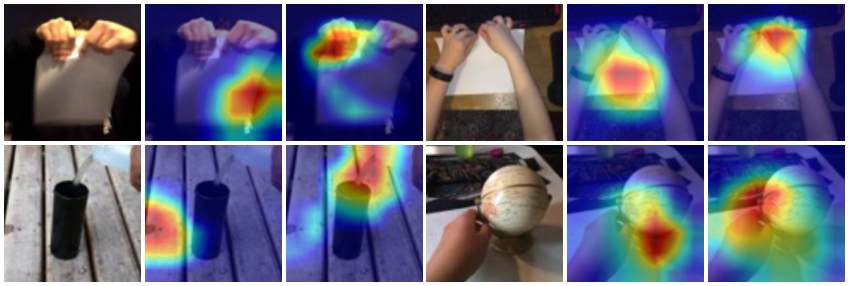

Figure 3: Visualization of activation maps with CAM. *Left*: video, *Middle*: baseline, *Right*: TDN.

Study on long-term TDM. In this study, 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 1b. 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 to enhance discriminative features across different dimensions. It is observed that our proposed spatiotemporal attention form of L-TDM is better than no attention (51.3% vs. 44.1%) and channel attention (51.3% vs. 50.9%). Then, we investigate the effectiveness of multi-scale architecture in difference feature extraction and it is able to improve performance form 49.7% to 51.3%, 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 1.3%.

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 1c. We first report the performance of baseline of without S-TDM or L-TDM, namely only with 1D temporal convolutions in latter stages for temporal modeling, and its accuracy is 45.2%. Then we separately add S-TDM and L-TDM into the baseline, and they obtain the performance of 49.5% and 48.9%. 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 51.3%, 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 1d. For fair comparison, these methods all use the ResNet50 as backbones and 8 frames as input, so their FLOPs are similar to each other. We first compare with TSN baseline (Wang et al., 2016) only with temporal fusion at the score level and temporal convolution (Tran et al., 2018) placed in each ResNet block. We find that the performance of our TDN is much better than those baselines. We also compare with the recent temporal modeling methods, such as TSM (Lin et al., 2019), TEINet (Liu et al., 2020), TEA (Li et al., 2020), and TAM (Fan et al., 2019). These modules are designed to efficiently model temporal information based on 2D CNNs, and we observe that our proposed TDM outperforms them by an evident improvement.

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 section, we focus on comparing 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 in this section. As for backbones, it is well known using deeper models could contribute to higher performance, but this is not the contribution of our method. So, we keep the backbone as ResNet50 and compare with previous methods with similar backbones. We also list the best performance of previous approaches with more powerful backbones for reference.

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 (Wang et al., 2016) and TRN (Zhou

Method	Backbone	Frames	FLOPs	Sth-Sth v1		Sth-Sth v2	
				Top1	Top5	Top1	Top5
TSN-RGB (Wang et al., 2016)	BNInception	8	16G	19.5%	-	-	-
TRN-Multiscale (Zhou et al., 2018)	BNInception	8	33G	34.4%	-	48.8%	77.6%
S3D-G (Xie et al., 2018)	Inception	64	71.38G	48.2%	78.7%	-	-
TSM (Lin et al., 2019)	ResNet50	16	65G	47.2%	77.1%	-	-
TEINet (Liu et al., 2020)	ResNet50	16	66G	49.9%	-	62.1%	-
TEA (Li et al., 2020)	ResNet50	16	70G	51.9%	80.3%	-	-
TAM (Fan et al., 2019)	bLResNet50	16×2	47.7G	48.4%	78.8%	61.7%	88.1%
ECO (Zolfaghari et al., 2018)	BNIncep+Res18	16	64G	41.6%	-	-	-
ECO$_{En}$Lite (Zolfaghari et al., 2018)	BNIncep+Res18	92	267G	46.4%	-	-	-
I3D (Carreira & Zisserman, 2017)	ResNet50	32×2	306G	41.6%	72.2%	-	-
NL I3D (Wang & Gupta, 2018)	ResNet50	32×2	334G	44.4%	76.0%	-	-
NL I3D+GCN (Wang & Gupta, 2018)	ResNet50+GCN	32×2	606G	46.1%	76.8%	-	-
TDN	ResNet50	8	36G	51.3%	79.3%	63.1%	88.4%
TDN	ResNet50	16	72G	52.1%	80.5%	64.4%	89.4%
TDN	ResNet50	8+16	108G	**54.1%**	**82.4%**	**66.1%**	**90.3%**
TAM (Fan et al., 2019)	bLResNet101	32×2	128.6G	53.1%	82.9%	65.2%	90.3%

Table 2: Comparison with the state-of-the-art methods on **Something-Something V1 and V2**.

Method	Backbone	Frames×Clips×Crops	FLOPs×views	Top1	Top5
TSN (Wang et al., 2016)	InceptionV3	25×1×10	3.2G×250	72.5%	90.2%
S3D-G (Xie et al., 2018)	InceptionV1	64×10×3	71.4G×30	74.7%	93.4%
R(2+1)D (Tran et al., 2018)	ResNet34	32×10×1	152G×10	74.3%	91.4%
TSM (Lin et al., 2019)	ResNet50	16×10×3	65G×30	74.7%	91.4%
TEINet (Liu et al., 2020)	ResNet50	16×10×3	66G×30	76.2%	92.5%
TEA (Li et al., 2020)	ResNet50	16×10×3	70G×30	76.1%	92.5%
TAM (Fan et al., 2019)	bLResNet50	48×3×3	93.4G×9	73.5%	91.2%
ARTNet (Wang et al., 2018a)	ResNet18	16 ×25×10	23.5G×250	70.7%	89.3%
I3D (Carreira & Zisserman, 2017)	InceptionV1	64×N/A×N/A	108G×N/A	72.1%	90.3%
NL I3D (Wang et al., 2018b)	ResNet50	128×10×3	282G×30	76.5%	92.6%
SlowOnly (Feichtenhofer et al., 2019)	ResNet50	8×10×3	41.9G×30	74.8%	91.6%
SlowFast (Feichtenhofer et al., 2019)	ResNet50	(8+64)×10×3	65.7G×30	77.0%	92.6%
TDN	ResNet50	8×10×3	36G×30	76.6%	92.8%
TDN	ResNet50	16×10×3	72G×30	77.5%	93.2%
TDN	ResNet50	(8+16)×10×3	108G×30	**78.4%**	**93.6%**
SlowFast+NL (Feichtenhofer et al., 2019)	ResNet101	(16+128)×10×3	234G×30	79.8%	93.9%

Table 3: Comparison with the state-of-the-art methods on **Kinetics-400**.

et al., 2018), 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 (Xie et al., 2018), R(2+1)D (Tran et al., 2018), TSM (Lin et al., 2019), TEINet (Liu et al., 2020), and TAM (Fan et al., 2019), and our TDN consistently outperforms them on both datasets, demonstrating the effectiveness of TDM in temporal modeling for action recognition. Finally, we compare with more recent 3D CNNs based methods, such as I3D (Carreira & Zisserman, 2017), Non-local I3D (Wang et al., 2018b), and SlowFast (Feichtenhofer et al., 2019), and our TDN can still obtain slightly better performance than those methods, with a relatively smaller computational cost. We also combine the results of 8-frame and 16-frame TDNs and it can further boost the performance on both datasets. Finally, for reference, we also provide the best result of previous methods with more powerful backbones and more frames.

4.4 Visualization of activation maps

We visualize the class activation maps with Grad-CAM (Zhou et al., 2016; Selvaraju et al., 2020) and results are shown in Figure 3. 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.

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 a temporal module (TDM), for capturing both short-term and long-term temporal information in a video. We present two specific and efficient 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.

REFERENCES

João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pp. 4724–4733, 2017.

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

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*, pp. 2625–2634, 2015.

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*, pp. 2261–2270, 2019.

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

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pp. 6201–6210, 2019.

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*, pp. 5843–5851, 2017.

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*, pp. 8401–8408, 2019.

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

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

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pp. 7132–7141, 2018.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. In *ICML*, pp. 495–502, 2010.

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

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

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.

Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. TEA: temporal excitation and aggregation for action recognition. *CoRR*, abs/2004.01398, 2020.

Ji Lin, Chuang Gan, and Song Han. TSM: temporal shift module for efficient video understanding. In *ICCV*, pp. 7082–7092, 2019.

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.

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*, pp. 4694–4702, 2015.

Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. *ICCV*, pp. 5534–5542, 2017.

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. *International Journal of Computer Vision*, 128(2):336–359, 2020.

Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. *CoRR*, abs/2001.06499, 2020.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pp. 568–576, 2014.

Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pp. 4489–4497. IEEE Computer Society, 2015.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pp. 6450–6459, 2018.

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.

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*, volume 9912, pp. 20–36. Springer, 2016.

Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, pp. 1430–1439, 2018a.

Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, pp. 413–431, 2018.

Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pp. 7794–7803. IEEE Computer Society, 2018b.

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, pp. 318–335. Springer, 2018.

Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R Scott, and Limin Wang. V4d: 4d convolutional neural networks for video-level representation learning. In *International Conference on Learning Representations*, 2019.

Yue Zhao, Yuanjun Xiong, and Dahua Lin. Recognize actions by disentangling components of dynamics. In *CVPR*, pp. 6566–6575, 2018.

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

Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, volume 11205, pp. 831–846. Springer, 2018.

Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: efficient convolutional network for online video understanding. In *ECCV*, pp. 713–730, 2018.

Method	Pretrain	Backbone	UCF101	HMDB51
TSN Wang et al. (2016)	ImageNet	Inception V2	86.4%	53.7%
P3D Qiu et al. (2017)	ImageNet	ResNet50	88.6%	-
C3D Tran et al. (2015)	Sports-1M	ResNet18	85.8%	54.9%
I3D Carreira & Zisserman (2017)	ImageNet+Kinetics	Inception V2	95.6%	74.8%
ARTNet Wang et al. (2018a)	Kinetics	ResNet18	94.3%	70.9%
S3D Xie et al. (2018)	ImageNet+Kinetics	Inception V2	96.8%	75.9%
R(2+1)D Tran et al. (2018)	Kinetics	ResNet34	96.8%	74.5%
TSM Lin et al. (2019)	Kinetics	ResNet50	96.0%	73.2%
STM Jiang et al. (2019)	ImageNet + Kinetics	ResNet50	96.2%	72.2%
TEA Li et al. (2020)	ImageNet + Kinetics	ResNet50	96.9%	73.3%
TDN	ImageNet + Kinetics	ResNet50	**97.4%**	**76.3%**

Table 4: Comparison with the state-of-the-art methods on **UCF101 and HMDB51**.

A Results on the UCF101 and HMDB51

To further verify the generalization ability of TDN, we transfer the learned 16-frame TDN models from the Kinetics-400 dataset to the UCF101 and HMDB51. These two datasets are relatively small and the action recognition performance on them already saturates. We follow the standard evaluation scheme on these two datasets and report the mean accuracy over three splits. The results are summarized in Table 4. We compare our TDN with previous state-of-the-art methods such as 2D baselines of TSN Wang et al. (2016), 3D CNNs of I3D Carreira & Zisserman (2017) and C3D Tran et al. (2015), R(2+1)D Tran et al. (2018), and other temporal modeling methods Li et al. (2020); Jiang et al. (2019). From the results, we can see that our TDN is able to outperform these methods, and the performance improvement is more evident on the dataset of HMDB51 by around 2.5%. The action classes in HMDB51 are more relevant with motion information, and thus temporal modeling is more important on this dataset.

B Ablation study on stages of S-TDM and L-TDM

S-TDM	L-TDM	FLOPs	Top1
-	-	33G	45.2%
Stage 1	Stage 2-5	35G	49.9%
Stage 1-2	Stage 3-5	36G	**51.3%**
Stage 1-3	Stage 4-5	38G	50.8%

Table 5: Ablation study of S-TDM and L-TDM on Something-Something V1.

We further perform ablation study on which stage to use short-term TDM (S-TDM) or long-term TDM (L-TDM) and the results are shown in Table 5. From these results, we see that adding more S-TDMs into the main network will increase the network computational cost slightly. The setting of using S-TDM in stages 1-2 and L-TDM in stages 3-5 obtains the best performance.

C Running time analysis

Method	Frames×Clips×Crops	Time (ms/video)	Top1 (%)
TSN	$8 \times 1 \times 1$	7.9	19.7
TSM	$16 \times 1 \times 1$	16.7	47.2
STM	$8 \times 1 \times 1$	11.1	47.5
I3D	$32 \times 3 \times 2$	2095	41.6
S-TDM	$8 \times 1 \times 1$	12.3	49.5
L-TDM	$8 \times 1 \times 1$	15.8	48.9
TDN	$8 \times 1 \times 1$	22.1	51.3

Table 6: Running time analysis on a Tesla V100.

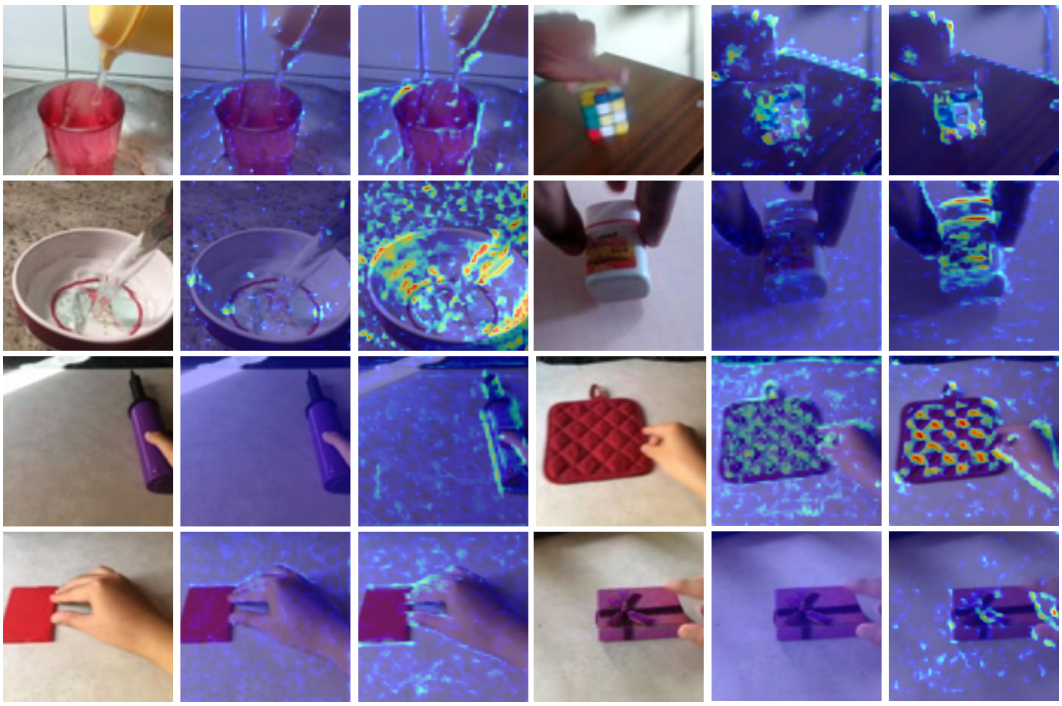

Figure 4: Visualization of Res2 features with Grad-CAM. We use 8-frame TDN models to visualize on the Something-Something V1 dataset. Left: video, Middle: baseline, Right: TDN with S-TDM. Note that we only show visualization on the center frame of sampled 8 frames.

We report the inference time of our TDN with on Tesla V100 as follows. The testing batchsize is set as 16 and the running time include all evaluation, including loading data and network inference. The results are reported in Table 6. From these results, we see that our TDN is slower than previous method but still could run in real-time (i.e. $\geq$25 FPS).

D VISUALIZATION ANALYSIS

To further investigate the performance the TDN models, we use the technique of Grad-CAM Selvaraju et al. (2020) to visualize the feature representation of different models. Specifically, to better understand the effect of short-term TDM, we visualize the the features in Res2 stage of baseline model (corresponding to the first row in Table 1(c) of main article) and the TDM model only with S-TDM (corresponding to third row in in Table 1(c) of main article), and the results are shown in Figure 4. Note that, these visualizations only are performed on the center frame of 8-frame models. From these results, the models equipped with S-TDM focuses more on motion-relevant information. Then, we give more visualization examples of activation maps in Figure 5 and Figure 6. In these results, we give the visualization results on 8 frames and compare our TDM models with the baseline method (corresponding to the first row in Table 1(c) of main article). We could see that our TDN is able to yield more reasonable class activation maps than the baseline method.

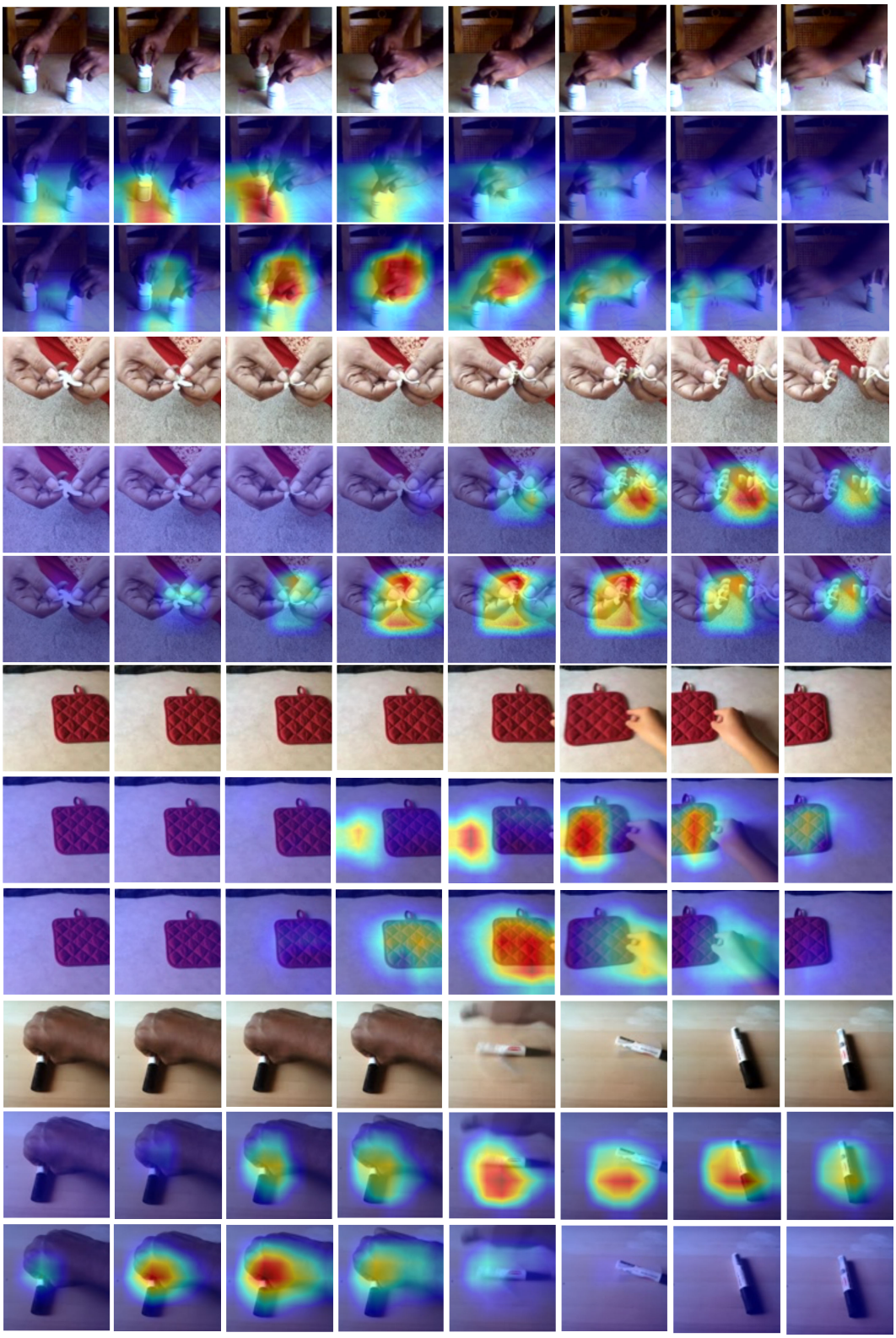

Figure 5: Visualization of activation maps with Grad-CAM. We use 8-frame TDN models to visualize on the Something-Something V1 dataset. In the first row, we plot the 8 RGB frames. In the second row, we plot the activation maps of the baseline method without temporal difference module (TDM). In the third row, we plot the activation maps of the TDN models.

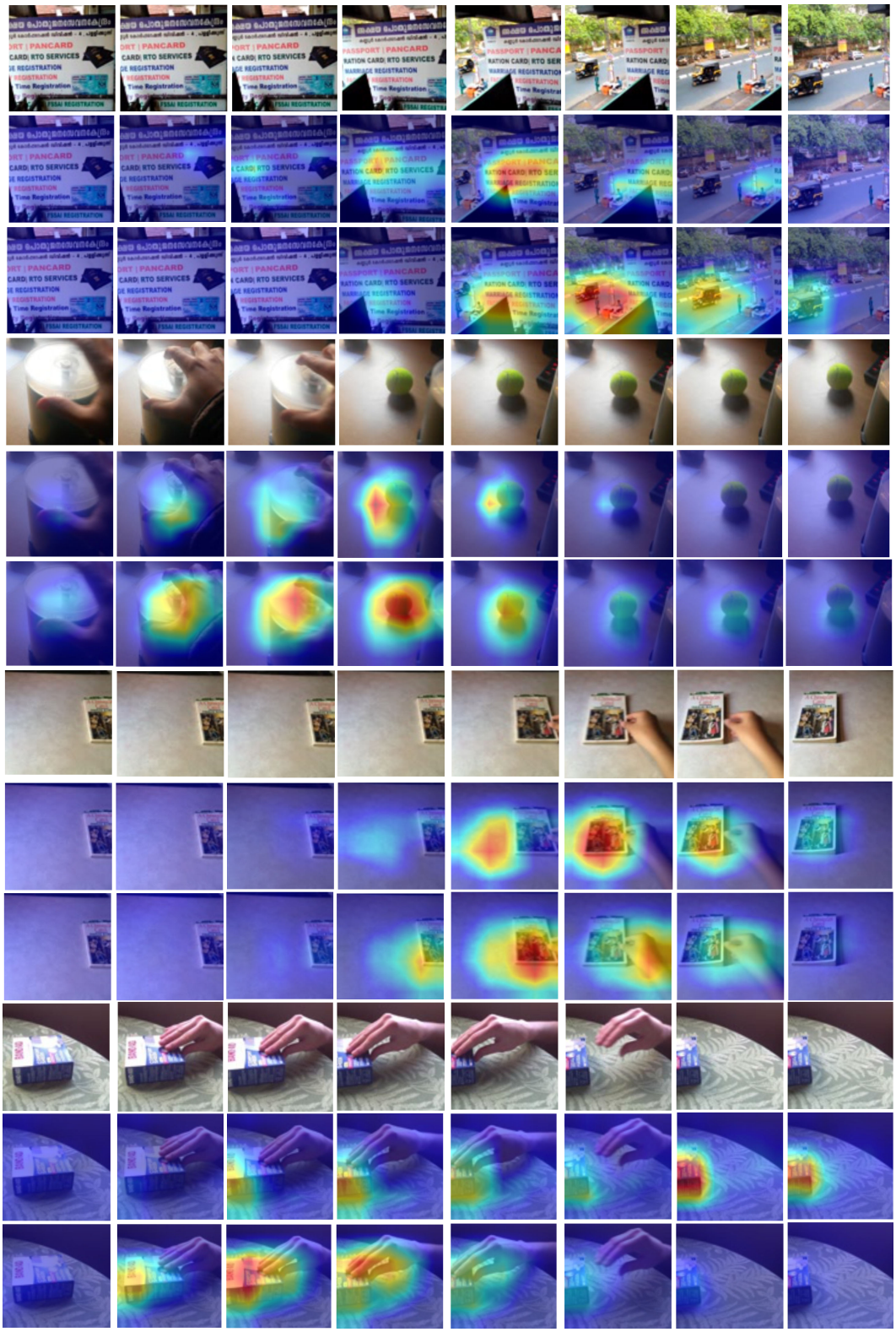

Figure 6: Visualization of activation maps with Grad-CAM. We use 8-frame TDN models to visualize on the Something-Something V1 dataset. In the first row, we plot the 8 RGB frames. In the second row, we plot the activation maps of the baseline method without temporal difference module (TDM). In the third row, we plot the activation maps of the TDN models.