TRM:Temporal Relocation Module for Video Recognition

Yijun Qian, Guoliang Kang, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann
Language Technologies Institute, Carnegie Mellon University
yijunqian@cmu.edu, kgl.prml@gmail.com, lijun@cmu.edu, {wenhel, alex}@cs.cmu.edu

Abstract

One of the key differences between video and image understanding lies in how to model the temporal information. Due to the limit of convolution kernel size, most previous methods try to model long-term temporal information via sequentially stacked convolution layers. Such conventional manner doesn't explicitly differentiate regions/pixels with various temporal receptive requirements and may suffer from temporal information distortion. In this paper, we propose a novel Temporal Relocation Module (TRM), which can capture the long-term temporal dependence in a spatial-aware manner adaptively. Specifically, it relocates the spatial features along the temporal dimension, through which an adaptive temporal receptive field is aligned to each pixel spatial-wisely. As the relocation is performed within the global temporal interval of input video, TRM can potentially model the long-term temporal information with an equivalent receptive field of the entire video. Experiment results on three representative video recognition benchmarks demonstrate TRM outperforms previous state-of-the-arts noticeably and verifies the effectiveness of our method.

1. Introduction

Video understanding is an important computer vision task and has been adopted in various scenarios [12, 2, 7, 6, 16, 31]. The recent success of video understanding can be primarily attributed to the advancement of temporal modeling. However, it remains challenging to effectively aggregate temporal information, especially for distinguishing activities with various temporal lengths and complex spatial-temporal contexts. In the former works, different algorithms have been proposed for temporal information aggregation. A series of works [18, 27, 32, 46] are established on the two-stream 2D CNNs. In such framework, a separate stream, which relies on extra temporal features (*e.g.* optical flow), is employed to incorporate the temporal information.

Another line of works [28, 35, 15, 6, 7, 12, 30, 21, 36] resort to 3D convolution networks to model spatial and temporal information simultaneously. The spatial-temporal receptive field is progressively enlarged via stacking the 3D local convolution kernels.

Different from all previous lines, Temporal Shift Module (TSM) [22] provides a new perspective for aggregating the temporal information. Via a parameter-free temporally shift module, TSM aggregates the temporal information efficiently with only 2D convolutions. However, since the temporal shift module is manually designed, the shift offsets are restricted in a small fixed temporal range and cannot be adaptively adjusted during training or inference.

In summary, previous works sequentially stack local temporal/spatial-temporal convolutions to model long-term temporal information that may suffer from the temporal information distortion. Moreover, they perform temporal aggregation in a spatial-agnostic way, ignoring the fact that the optimal temporal receptive field may vary dramatically for different regions of a frame. Although there are recent works like ACTION-Net [33] which improved TSM through adding attention modules, these methods still followed the temporal shift operation and didn't enhance the long-term spatial-temporal modeling capability within each operation.

In this paper, we propose a novel Temporal Relocation Module (TRM). TRM aims to relocate the spatial features along the temporal dimension to enable long-term temporal modeling in a spatial-aware manner. Specifically, TRM applies convolution kernels on the temporal-channel dimensions to determine the relocated locations of each pixel. Through a linear sampling function, such relocation module can be trained in an end-to-end schema. TRM can be implemented with 2D CNNs to enable the subsequent convolution layers the capability of performing spatial-temporal aggregation with temporal relocated features. As the relocation is performed within the whole temporal interval of input, TRM can model the long-term temporal information with an equivalent global temporal receptive field within each relocation operation. Moreover, the learned temporal relocation values are pixel-wise and adaptive according to input videos.

In a nutshell, the contributions of our paper are summa-

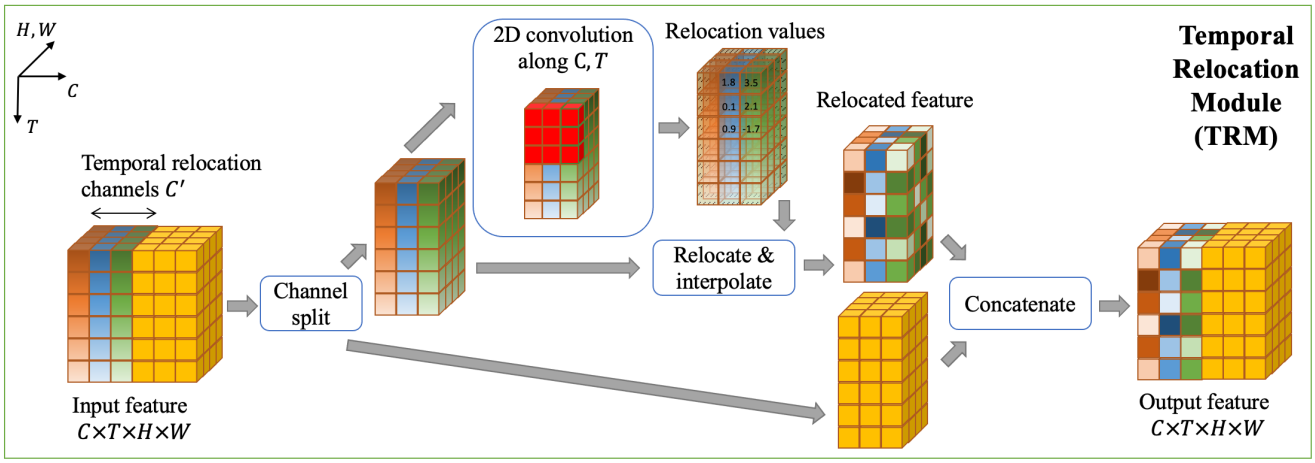

Figure 1. The Illustration of Temporal Relocation Module. For visualization, we merged the H and W dimensions to show the 4D input feature. Each color represents a specific channel. The features are relocated within each channel on the temporal dimension pixel-wisely according to the learned relocation matrix.

rized as follows:

- We propose a new perspective of modeling temporal information through temporal relocation operation. It enables 2D CNNs model spatial-temporal information with a global temporal receptive field and enhances the long-term temporal modeling capability.
- We propose Temporal Relocation Module (TRM), which takes the difference of optimal temporal receptive among pixels/regions into account and enables the temporal receptive field for each location to be determined adaptively through temporal relocation operations.
- Experiment results on three video recognition benchmarks (*i.e.* Kinetics, something-something V2, and HMDB51) demonstrate the superiority of our method.

2. Related Work

Two-stream 2D CNN These methods operate convolution independently over the temporal dimension and resort to temporal motion information like optical flow or RGB diff of adjacent frames for temporal modeling [18, 27, 32, 46, 25]. The usage of motion information makes them different from image-level understanding works[38, 24, 39, 37, 42]. For example, Wang proposes a framework for video-based action recognition with parallel spatial convolution network and temporal convolution network [32]. Karen Simonyan and Andrew Zisserman propose a two-stream convolution architecture that incorporates spatial and temporal networks through the usage of optical flow.[27] As we mentioned before, such methods contain fewer parameters and are easier for training. However, the motion information is extracted through adjacent images and can not represent long-term temporal information.

3D CNN Different from 2D CNN, 3D CNN can extract temporal and spatial information simultaneously. Works[15, 14, 6, 7] with 3D CNNs have achieved state-of-the-art (SOTA) results on many data sets, especially after the release of large video action data sets like Kinetics[19] and Activity Net[1]. For example, Du proposes a simple, yet effective approach for spatio-temporal feature learning using deep 3D CNN[28]. Carreira implements a two-stream inflated 3D convolution network [2]. Through stacked 3D convolutions, 3D CNN can capture long-range temporal relationships. However, given set temporal kernel size, it can only extract temporal information from frames within a certain temporal window once. What's more, 3D CNNs contain many more parameters and make them more difficult to train.

Temporal Modeling Through 1D Convolution Given the drawbacks of traditional 2D CNN and 3D CNN mentioned above, Du proposed R(2+1)D which factorizes the 3D convolutional filters into 2D spatial convolution kernel and 1D temporal convolution kernel[30]. Although TRM also implements convolution kernels for temporal modeling, the convolution kernels are used for temporal relocation instead of aggregation. Furthermore, TRM has much larger temporal receptive field and considers the temporal receptive requirements of different regions. The experiments in the ablation study section show the improvements brought by integrating TRM as an adaptive temporal receptive extension module.

Temporal Shift Operation Different from other meth-

ods, temporal shift module does not contain convolution operation and can be inserted easily into 2D CNNs to achieve temporal modeling at zero computation and zero parameters. Compared with our work, their temporal shift module only moves the feature map along the temporal dimension one frame forward or backward. Although it is computationally free, the shift distance is a fixed parameter which can't get updated during training procedure and may make it not the optimal one for different kinds of videos. The extension works[44, 10, 26] make the temporal shift values learnable and unifies the shift operation in temporal and spatial realm. But, they did not take the temporal receptive requirements of different regions into consideration. Meanwhile, RubiksNet[10] sets the temporal shift values as free parameters. Although it is learnable in training procedure, it won't update according to the input videos during inference. The temporal relocation values learned by TRM, however, is extracted from the feature map through convolution kernels and can be updated adaptively according to the input video during inference. In other words, RubiksNet learns a "3D CNN" for each channel and TRM learns how to adaptively combine a group of "3D CNNs" for input video.

3. Temporal Relocation Module (TRM)

In this section, we will firstly illustrate temporal relocation operation. Then we will introduce the structure of TRM, which enables the spatial-aware temporal relocation values updated adaptively and end-to-end according to the loss back-propagated from the final target. Finally, we will introduce how to integrate TRM into video recognition methods.

3.1. Temporal Relocation Operation

The convolution operation consists two steps. Firstly, it samples candidates through a grid matrix G over the input features. Then the sampled candidates are multiplied with a weight matrix W and summed up. For example, for a 3×3 2D convolution kernel with stride set as 1, its grid matrix is: $G_{2D} = \begin{bmatrix} (-1,-1) & (-1,0) & (-1,1) \\ (0,-1) & (0,0) & (0,1) \\ (1,-1) & (1,0) & (1,1) \end{bmatrix}$. Given an input feature map X whose size is $C \times T \times H \times W$. C represents the number of channels, T represents the number of frames, and HW denotes the spatial resolution. Two-stream 2D CNN methods implement 2D CNNs frame by frame directly for video understanding. For simplicity, we use p to represent the location of a pixel on the spatial dimension (H and W). For each spatial location p_0 on the output feature map Y, the 2D convolution procedure is represented as:

$$Y(i, t_0, p_0) = \sum_{c=0}^{C} \sum_{p_n \in G^i_{2D}} W^i_{2D}(p_n) X(c, t_0, p_0 + p_n) \quad (1)$$

where W^i_{2D} represents the weight matrix of the i^{th} 2D convolution kernel and G^i_{2D} represents its grid matrix. p_n enumerates the locations in G^i_{2D}. $p_0 + p_n$ represents the sampled spatial location on input feature map X. These methods extract the features temporal independently. Therefore, they usually require extra temporal information, e.g. optical flow, for temporal modeling. However, the motion information only captures short-term changes and fails to extract long-term real temporal relationship. Later methods resort to 3D CNNs or factorizing it to a 2D spatial convolution and a 1D temporal convolution. For simplicity, we ignore the spatial convolution part and focuse on the temporal convolution operation. The 1D temporal convolution operation is derived as:

$$Y(i, t_0, p_0) = \sum_{c=0}^{C} \sum_{t_n \in G^i_{1D}} W^i_{1D}(t_n) X(c, t_0 + t_n, p_0) \quad (2)$$

Similar to the previous definition, W^i_{1D} represents the weight matrix of the i^{th} 1D convolution kernel, and G^i_{1D} represents its grid matrix for sampling. The size of kernels limits the temporal receptive field of each convolution operation. To model long-term temporal features, such local convolution kernels are stacked sequentially. However, such manner may suffer from the temporal information distortion and cannot satisfy the requirements of optimal temporal receptive filed of different videos. To solve this, we intend to release the limit of t_n and extend it to the length of the entire video clip. Under the intuition of such idea, we propose temporal relocation operation. It builds "shortcuts" for long-term temporal information through relocating the features on the temporal dimension. R represents the grid matrix of temporal relocation operation, whose size is $C \times T \times H \times W$. Each grid of the matrix represents the temporal relocation value of the corresponding pixel in feature map X. The grids of R range from $-T$ to T. When integrating R into the derivation of previous 1D temporal CNNs, the procedure turns to:

$$Y(i, t_0, p_0) = \sum_{c=0}^{C} \sum_{t_n \in G^i_{1D}} W^i_{1D}(t_n) X(c, t_0 + t_g + t_n, p_0) \quad (3)$$

$$t_g = R(c, t_0, p_0 + p_n) \quad (4)$$

As is shown in Figure 1, the temporal receptive field of the temporal relocation now extends to the equivalent length of the entire video. Meanwhile, each pixel can be aligned with the optimal temporal relocation value, which makes the model more robust compared with a unified temporal shit value. When adding R into the derivation of previous

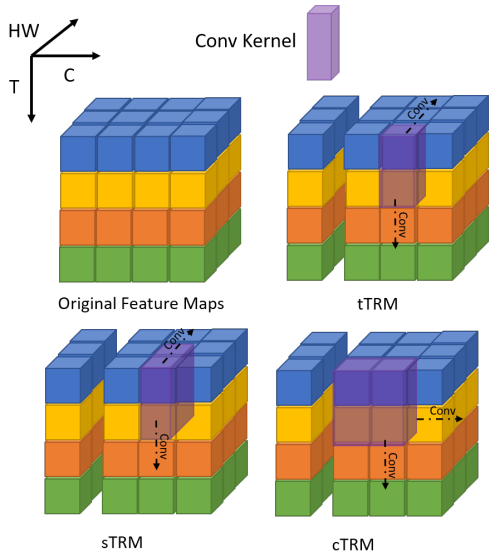

Figure 2. Illustration of different types of contextual information. tTRM only uses the temporal contextual information for relocation. sTRM resorts to the spatial-temporal contextual information and cTRM resorts to the channel-temporal contextual information.

2D CNNs, the procedure turns to:

$$Y(i,t_0,p_0) = \sum_{c=0}^{C} \sum_{p_n \in G_{2D}^i} W_{2D}^i(p_n) X(c, t_0 + t_g, p_0 + p_n) \tag{5}$$

Compared with previous derivation, the feature of p_0+p_n now comes from the pixel with the same spatial location in the same channel, but $|t_g|$ frames forward or backward. With the implementation of such relocation operation, the 2D convolution can model spatial-temporal information simultaneously with the global temporal receptive field. Meanwhile, temporal information between long-range frames does not need to be transmitted through stacked local convolution kernels for modeling.

3.2. Temporal Relocation Module

Unlike classification task with a clear target and criterion, it is difficult to evaluate the performance of relocation, (*i.e.* how well each pixel should be relocated). Thus, alternatively, we evaluate the performance of action recognition for our method. The grid matrix R is optimized according to the loss back-propagated from the cross-entropy function of classification and requires the whole procedure of temporal relocation derivable. Moreover, the optimal temporal receptive is varied for different actions and videos. Thus, we hope the model can make temporal relocation not only spatial-wisely but adaptively according to input videos as well. TRM uses local contextual information to predict the optimal temporal receptive field. We made an exploration of the dimension of contextual information. As is shown in Figure 2, we explored the usage of temporal contextual feature (tTRM), spatial-temporal contextual feature (sTRM) and channel-temporal contextual feature (cTRM). To preserve the integrity of spatial information, we only make temporal relocation operations on part of the channels. Suppose we select the first $C^{'}$ channels out of C channels.

$$R_t^{'}(i,t_0,p_0) = \sum_{t_n \in G_t^i} \sum_{c=0}^{C'} W_t^i(t_n) X(c, t_0 + t_n, p_0) \tag{6}$$

$$R_t = tanh(R_t^{'}) \times T \tag{7}$$

R_t represents the grid matrix of tTRM, G_t^i is the grid matrix of the i^{th} 1D temporal convolution kernel and W_t^i represents its weight matrix. A $\tanh$ function is implemented after the filters and the output is multiplied by T, the temporal length of input clip, to transfer the range of relocation values between -T and T. Similarly, we can derive the procedure of sTRM and cTRM.

$$R_s^{'}(i,t_0,p_0) = \sum_{(t_n,p_n) \in G_s^i} \sum_{c=0}^{C'} W_s^i(t_n,p_n) X(c, t_0{+}t_n, p_0{+}p_n) \tag{8}$$

$$R_s = tanh(R_s^{'}) \times T \tag{9}$$

$$R_c^{'}(c_0,t_0,i) = \sum_{(c_n,t_n) \in G_c^i} \sum_{s=0}^{HW} W_c^i(c_n,t_n) X(c_0{+}c_n, t_0{+}t_n, s) \tag{10}$$

$$R_c = tanh(R_c^{'}) \times T \tag{11}$$

R_s represents the grid matrix of sTRM, and R_c represents the grid matrix of cTRM. For both tTRM and sTRM, the kernels are reused by multiple pixels to extract temporal relocation values. However, in cTRM, each kernel only extracts the temporal relocation value for a specific spatial pixel. Although the temporal relocation values are all extracted spatial-wisely, the sharing of kernels may weaken the capability of learning spatial-specific features. According to our experiments, we found cTRM has the best performance, which is consistent with our assumption. The detailed results is presented in the ablation study part. To make the generated values derivable according to the loss back-propagated from relocated features, we can not use in-placement operations. The direct thought maybe through matrix multiplication. The problem is that we can easily translate the entire line or the entire column through matrix multiplication. However, it becomes challenging when need to translate the pixels independently. Suppose $Z_{c,t,h,w}$ the matrix to make all pixels besides $X(c,t,h,w)$ become

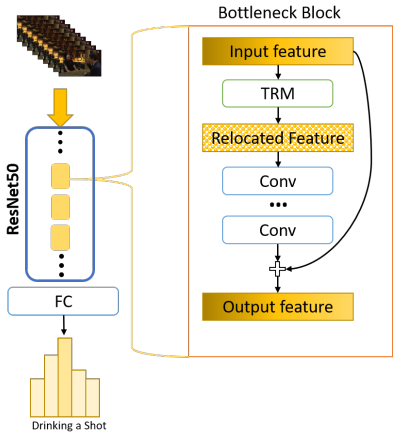

Figure 3. For each bottleneck block, a TRM module is inserted within the residual structure before the first convolution layer.

zero. $\mathrm{T}_{c,t,h,w}$ the matrix to translate pixel $\mathrm{X}(c,t,h,w)$ to $\mathrm{X}(c,t+R(c,t,h,w),h,w)$. The whole procedure can be represented as a series of matrix multiplication:

$$Y = \sum_{c,t,h,w} Z_{c,t,h,w} X T_{c,t,h,w} \tag{12}$$

c,t,h and w iterate on each degree. Apparently the calculation is too large for back-propagation. Given the inspiration from [4], we use a linear sampling function I to get the relocated value.

$$Y(c_0,t_0+t_g,p_0) = \sum_t I(t,t_0+t_g) X(c_0,t,p_0) \tag{13}$$

$$I(t,t_0+t_g) = max(0,1-|t-t_0-t_g|) \tag{14}$$

Where t iterates from -T to T. Thus, for each t_0, only two t will be non-zero. It makes the back-propagation much more efficient. In this way, the gradients are back-propagated continuously to temporal relocation values according to the final classification loss.

Discussions We can re-illustrate the operation of temporal shift (TSM) [22] from the perspective of temporal relocation. Temporal shift can be regarded as a special case of temporal relocation with manually designed relocation matrix R. TSM shifts all pixels of the first $1/8$ channels 1 frame forward and all pixels of the next $1/8$ channels 1 frame backward. It equals to setting the first $1/8$ channels of R as (1) , the next $1/8$ channels as (-1) and the rest elements as (0). The temporal relocation values are spatial-agnostic and can not be updated for different cases.

3.3. Implementation of Temporal Relocation Module in Backbone Networks

Since TRM preserves the shape of input features, it makes it easily inserted into current backbone networks. We illustrate the implementation in 2D ResNet50 as an example. The size of input feature maps determines the number of convolution kernels. So we only insert TRM in layer3 and layer4 of ResNet50. As is shown in Figure 3, for each bottleneck block, we insert one TRM module before the first convolution layer within the residual structure. ResNet50 is designed for image classification. Its input size is $B \times C \times H \times W$. For coding efficiency, we reshape the input video clip matrix to $BT \times C \times H \times W$. In each TRM module, it is reshaped back to $B \times C \times T \times H \times W$. The relocated features are concatenated with rest preserved channels and reshaped back to $BT \times C \times H \times W$ then forward to later 2D convolution layer.

4. Experiments

4.1. Datasets

Kinetics Kinetics[19] is a large video recognition benchmark. Presently the latest version contains approximately 650,000 video clips that cover 700 human action classes. For each action class, there are at least 600 video clips. To have an apple-to-apple comparison with the baseline method and other state-of-the-art models, we only use the version that covers 400 human action classes. The videos of kinetics are stored on Youtube. Some of the URL links have become invalid. Totally we successfully downloaded 222492 videos for training, 17545 for validation, and 34371 for testing.

Something Something V2 The 20BN-SOMETHING-SOMETHING dataset[13] is a large open-source human action data set. It contains 220,847 videos, 27,157 for testing, 24,777 for validation, and 168,913 for training. The data set covers 174 human-related actions. The annotations for training and validation set are released. The annotations of the test set are preserved hidden for official leaderboard ranking.

HMDB51 HMDB51[20] is a human motion benchmark. It contains 6,849 videos divided into 51 action categories, each contains a minimum of 101 clips. Followed the official split, we get 5,236 training clips and 1,530 testing clips.

4.2. Experimental Setup

Pretraining For the implementation of TRM in 2D CNNs, we choose the 2D ResNet50 as our backbone feature extraction network. Unless otherwise specified, the model is initialized with weights pretrained on ImageNet for all results reported on Kinetics. For the other benchmarks, the model is initialized with weights pretrained on Kinetics.

Data Pre-processing We followed the pre-processing procedure of [32]. The shorter side of raw images is resized

Figure 4. Visualization of the learned channel averaged temporal relocation matrix R given a video clip of drinking shots. Red regions represent the pixels assigned large temporal relocation values, and blue regions represent those aligned limited temporal relocation values. (For better visualization, we use the absolute value of learned temporal relocation grids)

to 256. The images are then cropped with scale-jittering and resized to 224×224. For experiments on Kinetics, we followed the dense-sampling procedure of [22] to generate 30 samples per video for inference. Meanwhile, to explore the capability of TRM on modeling long-term temporal information, we also followed the random-sampling procedure of [32] to sample 1 clip and 2 clips directly from the whole video then cropped it 10 times with the crop augmentation strategy in [32] to generate 10 and 20 samples per video for inference. For experiments on other benchmarks, the video is evenly separated into 8 segments. From each segment, we randomly selected 1 frame to generate the sampled clip for training and inference.

Training Details For Kinetics, the models are firstly trained 50 epochs on Kinetics without TRM, starting with a learning rate of 0.01. The learning rate drops to its 0.1 at 30, 40 epochs. Then we integrated TRM modules and re-train the model. The model is optimized with a starting learning rate of 0.001 for 30 epochs. The learning rate drops to its 0.1 at 13, 22, and 27 epochs. On something something V2, the model is optimized for 25 epochs, the learning rate starts as 0.001 and drops to its 0.1 at 10, 15, 20 epochs. On HMDB51, the model is trained 17 epochs with the learning rate starts as 0.001 and drops to its 0.1 at 7, 12, 15 epochs. For all the experiments, we optimize our model through SGD with momentum 0.9. Weight decay is set as 1e-4 for Kinetics and 5e-4 for the other two benchmarks.

4.3. Comparisons with the State-of-The-Arts

Kinetics As is shown in Table 1, the first compartment contains works based on 3D CNNs or mixture of 2D and 3D CNNs. The next compartment includes works based on 2D CNNs or (2+1)D CNNs. TRM is inferior to ir-CSN and ip-CSN. However, these two methods implement a much heavier backbone and use longer clips for inference. TRM achieves comparable results with SlowFast and X3D but uses shorter clips and 2/3 samples per video for inference. Compared with baseline methods, TRM surpasses TSM with 0.9% rank-1 accuracy with only 1/3 samples per video for inference and 1.3% rank-1 accuracy with 2/3 samples per video. Although TSM is more efficient (33G FLOPs vs 44G FLOPs), TRM has better long temporal modeling capability and requires fewer FLOPs per video (440G,880G vs 990G). In section 4.4, we did extra experiments to explore the improvements of modeling long-term temporal information brought by TRM.

Something Something V2 All the results in Table 2 besides TPN with ResNet-101 and PAN$_{Lite}$ only take 1 clip per video without crop augmentation nor clip augmentation for inference. According to the provided results, TRM is inferior to TPN+TSM, which implements much heavier 3D structures. TRM achieves comparable results against PAN$_{Lite}$ on the validation set. However, PAN$_{Lite}$ uses five times samples for inference. When compared with TSM, a special case of TRM, TRM outperforms it with 1.4% improvements on the validation set.

HMDB51 We compare our model with multiple state-of-the-arts on HMDB51. The results mentioned in Table 3 are coming from TSN[32], I3D[2], R(2+1)D[30], TSM[22], HATNet[9], RubiksNet[10] and STC[8]. According to the results, TRM outperforms the 2D baseline TSM with 2%. TRM even surpasses I3D with heavy 3D ResNet-101.

4.4. Ablation Studies

Long-term Temporal Information Modeling To evaluate the improvements in modeling long-term temporal information, we compare TRM with two counterparts by enlarging the temporal interval of the input video clip. We study performance of different methods under two sampling strategies. *The dense sampling strategy* separates the entire video into 10 splits. For each split, certain frames are randomly selected to form an input clip. For a single video sample, the model will have 10 clips for inference and the

Model	Backbone	Frames × Crops × Clips	Acc-1	Acc-5
I3D[2]	Inception V1	64×N/A×N/A	72.1	90.3
ECO-RGB[5]	BNIncep+3D ResNet-18	92×1×1	70.0	N/A
NL I3D[2]	3D ResNet-101	32×6×10	77.7	93.3
SlowFast[12]	3D ResNet-50	(16+4)×3×10	75.6	92.1
X3D[11]	3D ResNet-50	16×3×10	76.0	92.3
TPN[36]	3D ResNet-50	32*2×3×10	77.7	N/A
TDN[31]	ResNet-101	16×3×10	78.5	93.9
ir-CSN[29]	ResNet-101	32×3×10	76.2	92.2
ip-CSN[29]	ResNet-101	32×3×10	76.7	92.3
S3D-G[34]	Inception	64×N/A×N/A	74.7	93.4
TPN[36]	ResNet-50	8×10×1	73.5	N/A
TSN[32]	Inception V3	25×10×1	72.5	90.2
R(2+1)D[30]	ResNet-34	32 × 1 × 10	72.0	90.0
TEA[21]	ResNet-50	8 × 3 × 10	75.0	91.8
TSM[22]	ResNet-50	8×3×10	74.1	91.2
STM[17]	ResNet-50	16×3×10	73.7	91.6
TRM	ResNet-50	8×10×1	75.0	91.9
		8×10×2	75.4	92.3
		8×3×10	75.7	92.4

Table 1. Comparing with the state-of-the-arts on the validation set of Kinetics 400. Accuracy is measured on the validation set to make fair comparison with previously released works. N/A represents the number not provided in original publication.

Model	Backbone	Frames	Top1@Val
C3D[28]	VGG16	60	47.7
TRG[45]	ResNet-50	8	53.8
PAN$_{Lite}$[43]	ResNet-50+TSM	8+8×4	60.8
TSN[32]	ResNet-50	8	30.0
TPN+TSM[36]	ResNet-50	8	62.0
TRN[46]	Inception	8	50.8
TSM[22]	ResNet-50	8	59.1
RubiksNet[10]	ResNet-50	8	59.0
TRM	ResNet-50	8	61.1

Table 2. Comparing with the state-of-the-arts on Something Something V2.

Model	Backbone	Pretraining	Acc-1
TSN	ResNet-50	ImageNet	53.7
I3D	ResNet-101	Kinetics+ImageNet	74.8
R(2+1)D	ResNet-34	Kinetics	74.5
TSM	ResNet-50	Kinetics	73.7
HATNet	ResNet-50	HVU	73.4
STC	ResNet-50	Kinetics	74.9
RubiksNet	ResNet-50	Kinetics	74.6
TRM	ResNet-50	Kinetics	75.7

Table 3. Comparing with the state-of-the-arts on HMDB51.

final predictions are averaged over all clips. The temporal receptive field of each clip is only 1/10 of the entire video.

Model	Sampling	Acc-1	Acc-5
TSM[22]	Dense	74.1	91.2
	Sparse	71.2	88.3
TEA[21]	Dense	75.0	91.8
	Sparse	72.5	90.4
TRM	Dense	75.7	92.4
	Sparse	74.2	91.8

Table 4. Drop brought by sampling strategies on Kinetics 400.

For *the sparse sampling strategy*, frames are directly sampled from the video. The video is evenly separated into N segments. 1 frame is randomly selected from each segment to form the input clip. For an input video, the model only gets 1 clip for inference whose temporal receptive field equals to the entire video clip. Results in Table 4 show that our method consistently outperforms previous state-of-the-art and baselines for both sampling strategies. Interestingly, when transferring from dense sampling to sparse sampling, previous methods show much more distinct performance drop, *e.g.* for TSM, both the rank-1 and rank-5 accuracy drop 2.9%, and for TEA, the rank-1 drops 2.5% and rank-5 accuracy drops 1.4%. In contrast, TRM is more robust to such temporal receptive length change, the rank-1 and rank-5 accuracy only drop 1.5% and 0.6% respectively. The result further demonstrates the priority of TRM in modeling long-term temporal information.

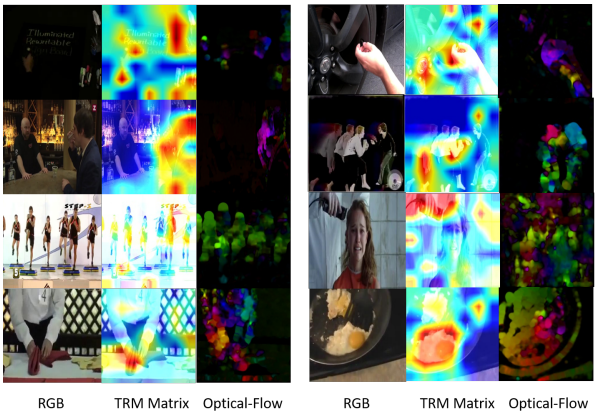

Figure 5. Visualization of the top 8 actions that TRM outperforms the spatial-agnostic one. It shows TRM can assign larger temporal relocation values to action related temporal valuable regions.

Effects of Spatial-aware Relocation To study the function of learning temporal relocation values pixel-wisely, we specially trained a spatial-agnostic TRM module for comparison on Kinetics. The spatial-agnostic TRM module adds an adaptive pooling layer after generating the temporal relocation matrix R. The pooling layer averages the values on the spatial dimension. Thus, all pixels in a specific channel share the same temporal relocation value. According to the results in Table 5, spatial-agnostic TRM drops 1.1% rank-1 accuracy on Kinetics. To better understand the impact caused by spatial-aware designing, we calculate the action-wise accuracy. The top 10 actions that TRM significantly out-performs are writing, drinking shots, doing aerobics, folding napkins, checking tires, swing dancing, shaving head, cooking egg, bending back, and juggling balls. According to the visualized examples shown in Figure 5, the temporal value varies greatly among different regions for these actions. Writing, for example, the region of moving pen contains the most valuable temporal information for recognition and requires large temporal receptive field to understand the whole procedure. For the other temporal irrelevant regions, limited temporal receptive is enough and the integrity of spatial information may be more important. The visualization of learned temporal relocation matrix illustrated in Figure 4 and Figure 5 demonstrate our module can learn such temporal valuable regions and assign them larger temporal offset values. Meanwhile, TRM is not the equivalent of an optical flow module. As shown in the case of folding napkins and shaving head, TRM aligns larger temporal receptive field to action-related temporal valuable regions, not just dynamic regions.

Model	Acc-1	Acc-5
TRM	75.4	92.3
spatial-agnostic TRM	74.6	91.5

Table 5. Improvements brought by spatial-aware temporal relocation.

Model	Acc-1	Acc-5
cTRM	75.4	92.3
sTRM	74.9	92.1
tTRM	74.6	91.8
TSM	74.1	91.2

Table 6. Difference brought by the selection of contextual information. All results besides TSM are reported with 2 clips random-sampling strategy (20 clips) in [32]

Selection of Contextual Information According to the experiments, we found the selection of contextual information has a noticeable influence. As is shown in Table 6, cTRM outperformed sTRM and tTRM with noticeable margin. The experimental result is consistent with our assumption. Compared with tTRM, sTRM uses extra spatial contextual information which may overcome the outliers and align temporal relocation values more robust. For both tTRM and sTRM, the convolution kernels are reused by all spatial pixels. These kernels are updated according to the gradients back-propagated from relocated features. Although the learned temporal receptive values are spatial-aware, the average of gradients may weaken the capability of kernels to learn spatial-specific features.

5. Conclusion

We propose TRM, a temporal relocation model for action recognition which can be implemented in multiple tasks[40, 41, 23, 3]. It models temporal information through relocating features on the temporal dimension pixel-wisely and adaptively. TRM enables followed 2D CNNs to model spatial-temporal information with a global receptive field. The temporal relocation values of TRM satisfy the optimal temporal receptive requirements for different regions and can be adaptively updated according to input videos.

6. Acknowledgements

This research is supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00340. This research is supported in part through the financial assistance award 60NANB17D156 from U.S. Department of Commerce, National Institute of Standards and Technology. This project is funded in part by Carnegie Mellon University's Mobility21 National University Transportation Center, which is sponsored by the US Department of Transportation.

References

[1] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[3] Xiaojun Chang, Wenhe Liu, Po-Yao Huang, Changlin Li, Fengda Zhu, Mingfei Han, Mingjie Li, Mengyuan Ma, Siyi Hu, and Guoliang Kang. MMVG-INF-Etrol@ TRECVID 2019: Activities in Extended Video. In *TRECVID*, 2019.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[5] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6638–6646, 2017.

[6] Ali Diba, Mohsen Fayyaz, Vivek Sharma, A Hossein Karami, M Mahdi Arzani, Rahman Yousefzadeh, and Luc Van Gool. Temporal 3d convnets using temporal transition layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1117–1121, 2018.

[7] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Amir Hossein Karami, Mohammad Mahdi Arzani, Rahman Yousefzadeh, and Luc Van Gool. Temporal 3d convnets: New architecture and transfer learning for video classification. *arXiv preprint arXiv:1711.08200*, 2017.

[8] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018.

[9] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Manohar Paluri, Jurgen Gall, Rainer Stiefelhagen, and Luc Van Gool. Holistic large scale video understanding. *arXiv preprint arXiv:1904.11451*, 2019.

[10] Linxi Fan, Shyamal Buch, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. Rubiksnet: Learnable 3d-shift for efficient video action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[11] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019.

[13] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, volume 1, page 5, 2017.

[14] Michelle Guo, Edward Chou, De-An Huang, Shuran Song, Serena Yeung, and Li Fei-Fei. Neural graph matching networks for fewshot 3d action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 653–669, 2018.

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[16] Yanli Ji, Feixiang Xu, Yang Yang, Ning Xie, Heng Tao Shen, and Tatsuya Harada. Attention transfer (ant) network for view-invariant action recognition. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 574–582, 2019.

[17] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2000–2009, 2019.

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[20] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[21] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020.

[22] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019.

[23] Wenhe Liu, Guoliang Kang, Po-Yao Huang, Xiaojun Chang, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, and Peng Chen. Argus: Efficient activity detection system for extended video analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 126–133, 2020.

[24] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 588–589, 2020.

[25] Yijun Qian, Lijun Yu, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. Adaptive feature aggregation for video object detection. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, pages 143–147, 2020.

[26] Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11966–11973, 2020.

[27] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[29] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5552–5561, 2019.

[30] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[31] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1895–1904, 2021.

[32] 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 *European conference on computer vision*, pages 20–36. Springer, 2016.

[33] Zhengwei Wang, Qi She, and Aljosa Smolic. Action-net: Multipath excitation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13214–13223, 2021.

[34] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 1(2):5, 2017.

[35] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.

[36] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020.

[37] Lijun Yu, Peng Chen, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. Training-free Monocular 3D Event Detection System for Traffic Surveillance. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3838–3843, Dec. 2019.

[38] Lijun Yu, Qianyu Feng, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann. Zero-virus: Zero-shot vehicle route understanding system for intelligent transportation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 594–595, 2020.

[39] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann. CMU Informedia at TRECVID 2020: Activity Detection with Dense Spatio-temporal Proposals. page 9.

[40] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann. CMU Informedia at TRECVID 2021: Activity Detection with Argus++.

[41] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. Argus++: Robust real-time activity detection for unconstrained video streams with overlapping cube proposals. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, 2022.

[42] Lijun Yu, Dawei Zhang, Xiangqun Chen, and Alexander Hauptmann. Traffic Danger Recognition With Surveillance Cameras Without Training Data. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Nov. 2018.

[43] Can Zhang, Yuexian Zou, Guang Chen, and Lei Gan. Pan: Towards fast action recognition via learning persistence of appearance. *arXiv preprint arXiv:2008.03462*, 2020.

[44] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. Token shift transformer for video classification. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 917–925, 2021.

[45] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *IEEE Transactions on Image Processing*, 29:5491–5506, 2020.

[46] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.