# EAC-Net: Efficient and Accurate Convolutional Network for Video Recognition

**Bowei Jin, Zhuo Xu**
iFLYTEK Research, Suzhou, China
{bwjin, zhuoxu}@iflytek.com

## Abstract

Research for computation-efficient video understanding is of great importance to real-world deployment. However, most of high-performance approaches are too computationally expensive for practical application. Though several efficiency oriented works are proposed, they inevitably suffer degradation of performance in terms of accuracy. In this paper, we explore a new architecture EAC-Net, enjoying both high efficiency and high performance. Specifically, we propose Motion Guided Temporal Encode (MGTE) blocks for temporal modeling, which exploits motion information and temporal relations among neighbor frames. EAC-Net is then constructed by inserting multiple MGTE blocks to common 2D CNNs. Furthermore, we proposed Atrous Temporal Encode (ATE) block for capturing long-term temporal relations at multiple time scales for further enhancing representation power of EAC-Net. Through experiments on Kinetics, our EAC-Nets achieved better results than TSM models with fewer FLOPs. With same 2D backbones, EAC-Nets outperformed Non-Local I3D counterparts by achieving higher accuracy with only about 7× fewer FLOPs. On Something-Something-V1 dataset, EAC-Net achieved 47% top-1 accuracy with 70G FLOPs which is 0.9% more accurate and 8× less FLOPs than that of Non-Local I3D+GCN.

## Introduction

Along with recent outstanding performance achieved by CNNs for image domains, such as semantic segmentation (Li et al. 2017; Long, Shelhamer, and Darrell 2015; Zhang et al. 2018) object detection (Liu et al. 2016; Ren et al. 2015) and classification (Simonyan and Zisserman 2014; He et al. 2016; Hu, Shen, and Sun 2018), the use of CNNs has been expanding significantly in other fields of computer vision. In particular, a variety of CNN based methods have been proposed for the tasks of video recognition, and have obtained better performance to methods by traditional handcrafted features (Laptev 2005; Laptev et al. 2008; Scovanner, Ali, and Shah 2007; Wang and Schmid 2013). Compared with still image classification, video recognition is a much more challenging task due to its higher dimensional and more complicated input signals.



For video classification task, recent popular architectures can be divided into two categories based on whether there is spatio-temporal fusion procedure. Two-stream 2D CNN (Simonyan and Zisserman 2014) as the representative for architectures without spatio-temporal fusion, capturing appearance and temporal information from RGB and optical flow inputs, which has turned out to be effective for video classification. 3D CNN (Ji et al. 2012; Tran et al. 2015) is the representative for models with spatio-temporal modeling, which have been experiencing rapid development since Kinetics dataset (Kay et al. 2017) available. Several recent works focus on modeling appearance information and inter-frame relations through separate branches within a spatio-temporal module (Zhou et al. 2018b; Qiu, Yao, and Mei 2017; Wang et al. 2018a) for empowering the representation for video tasks. Moreover, several other methods seek to alleviate training complexity of 3D kernels via factorization to facilitate learning of spatio-temporal information (Sun et al. 2015; Xie et al. 2018; Tran et al. 2018).

Most recent approaches focus their mind on how to learn expressive spatio-temporal representation for video classification task but all suffer the problem of heavy computation cost during inference. One factor leading to this problem is that extension of 2D CNN by integrating spatio-temporal modules usually results in large additional computation. Another factor is that most high-performance approaches tend to test on many clips, each of which is densely sampled from a raw video, and whole inference time becomes to multiply the time of once forward of a clip. Therefore, in practical use, studying approaches with better trade-off between accuracy and computation cost is very significant.

Given the aforementioned concerns, we propose our EAC-Net framework as one solution to the problem of trade-off between computation cost and performance. We designed MGTE block for temporal modeling, which consists of two branch, one is temporal max pooling which summarizes the responses between consecutive features and makes the temporal relation encoding become invariant to small changes. Another branch encodes the temporal relations of neighbor frames by temporal convolutions and then refined via a gate layer controlled by motion information. Inserting MGTE blocks into 2D CNNs makes the short-term spatio-

temporal information progressively and robustly injecting into the final representation. In addition, we propose ATE block for further processing features before feeding to classifier, which aims to extract temporal relations among video snippets at different time scales. We evaluate our method over Kinetics and Something-Something datasets. Experiment results show that compared with state-of-the-art approaches, our EAC-Net gives much better trade-off between accuracy and cost measured by FLOPs. For example, applying ResNet-50 as backbone, our EAC-Net achieves top-1 accuracy 75.3% at 196G FLOPS during inference. Besides, the learned representation is transferred to the Something V1 dataset and achieves accuracy of 47% at 150G FLOPs. The main contributions of our paper are summarized as follows:

- We propose a new generic building block, called MGTE block for temporal modeling, which exploits temporal relations and motion information among neighboring frames. It can be inserted into common 2D CNNs to empower the model to learn spatio-temporal features while adding less additional computational cost.
- The proposed ATE block applies series of temporal atrous 1D convolutions on the top of the backbone, further encoding the relations among video snippets, which is turned out to be effective in improving recognition performance.
- Based on ATE block and MGTE block, we propose a new architecture, EAC-Net, which outperforms some state-of-the-art approaches on both recognition accuracy and computational cost, showing better trade-off between precision and computational efficiency.

| Layer Name | C2D | Output Size $T \times S^2$ | EAC-Net | Output Size $T \times S^2$ |
|---|---|---|---|---|
| Conv1 | $1 \times 7^2, 1 \times 2^2, 64$ | $32 \times 112^2$ | $1 \times 7^2, 1 \times 2^2, 64$ | $32 \times 112^2$ |
| Pool1 | $1 \times 3^2, 1 \times 2^2, max$ | $32 \times 56^2$ | $1 \times 3^2, 1 \times 2^2, max$ | $32 \times 56^2$ |
| Res2 | $\begin{bmatrix} 1 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$ | $32 \times 56^2$ | $\begin{bmatrix} 1 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$ | $32 \times 56^2$ |
| Res3 | $\begin{bmatrix} 1 \times 1^2, 128 \\ 1 \times 3^2, 128 \\ 1 \times 1^2, 512 \end{bmatrix} \times 4$ | $32 \times 56^2$ | $\begin{bmatrix} 1 \times 1^2, 128 \\ 1 \times 3^2, 128 \\ 1 \times 1^2, 512 \end{bmatrix} \times 4$ | $32 \times 28^2$ |
| Res4_1 | $\begin{bmatrix} 1 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 1$ | $32 \times 14^2$ | $\begin{bmatrix} 1 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 1$ | $32 \times 14^2$ |
| Temporal Block | None | $32 \times 14^2$ | MGTE block (d=128) | $16 \times 14^2$ |
| Res4_{2:6} | $\begin{bmatrix} 1 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 5$ | $32 \times 14^2$ | $\begin{bmatrix} 1 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 5$ | $16 \times 14^2$ |
| Res5_1 | $\begin{bmatrix} 1 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 1$ | $32 \times 7^2$ | $\begin{bmatrix} 1 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 1$ | $16 \times 7^2$ |
| Temporal Block | None | $32 \times 7^2$ | MGTE block (d=256) | $8 \times 7^2$ |
| Res5_{2:3} | $\begin{bmatrix} 1 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 2$ | $32 \times 7^2$ | $\begin{bmatrix} 1 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 2$ | $8 \times 7^2$ |
| Head | GAP+Classifier | $1 \times 1^2$ | ATE block+Classifier | $1 \times 1^2$ |

Table 1: Architecture of C2D and EAC-Net model. The dimensions of kernels are denoted by $\{T \times S, \mathrm{C}\}$ for temporal, spatial, and channel sizes. The stride of each model layer can be represented as $\{$temporal stride, spatial stride$\}$. The input size is $32 \times 224 \times 224$. The backbone is ResNet-50. Here, d is a hyperparameter, denoting the dimension for reduction of input feature in a MGTE block.

## Related Work

On the contrary, 3D CNNs are naturally capable of learning spatio-temporal features from raw video frames. Recently, some successful variants of 3D CNNs are proposed and show their satisfied performance for video tasks on large scale datasets. I3D (Carreira and Zisserman 2017), inflated from 2D networks, enables to utilize pre-trained 2D CNNs to initialize 3D kernels, making a big progress on future studies of video tasks. Due to the fact that convolution layer extract features just in local neighborhood, non-local block (Wang et al. 2018b) is proposed to capture both spatial and temporal long-range dependencies and can be incorporated into inflated deep ResNet architectures for video classification. Most recently, CoST (Li et al. 2019) block is proposed to learn complementary features from multiple views and can be integrated into 2D ResNet architectures following the similar implementation of extending 2D ResNet to its C3D counterpart. Although the improvement on video classification by these methods cannot be negligible, they suffer from heavy computational cost in practical application. Then, some efforts are also made to reduce computation cost of 3D convolution such as S3D (Xie et al. 2018) and R(2+1)D (Tran et al. 2018) factorizing 3D convolution layer into a 2D spatial convolution and a 1D temporal convolution for reduction of computation. MF-Net (Chen et al. 2018b) utilize a model of small size, which is designed to have good trade-off between performance and computational speed. However, all these methods sample multiple subsets of the video densely and give the averaged output as the prediction, where model inference on each of the subsets is required, resulting a large amount of computation and making the methods difficult to deploy in practical application.

Our works are in some extent inspired by (Wang et al. 2018a; Zhou et al. 2018b; He et al. 2019), all of which learn spatio-temporal features by proposed blocks with multi-branch structures. Similarly, we designed the MGTE block in which two branches are employed for temporal modeling. Specifically, temporal max pooling is highly efficient in temporal modeling and thus chosen as one of the branches, whose encoding reserves majority of spatial information of neighboring frames, but without further exploration of temporal relations among frames. Motivated by this point, we add another computation-friendly branch for a supplement, which fully exploits motion information and temporal relations. Moreover, we propose ATE block to substitute the global pooling layer connected before classifier. This block aims to encode the complex relations among video snippets, which is hard to be processed trivially by global pooling operation. By zeroing out the responses from temporal relation

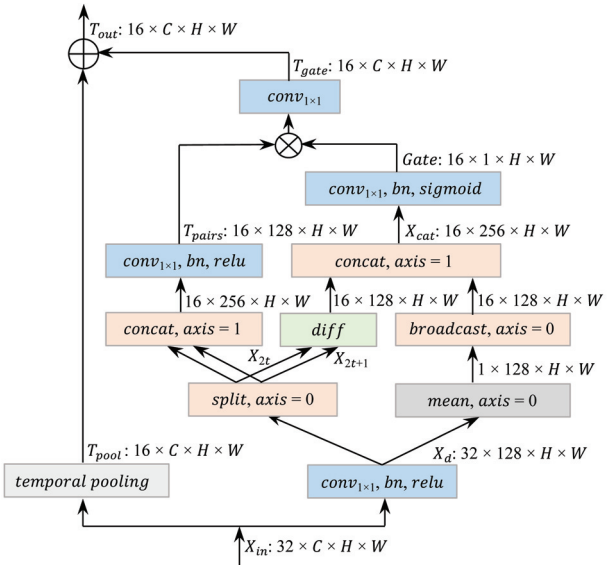


Figure 1: Architecture of MGTE block, where the feature maps are shown as the shape of their tensors, e.g., $X_{in} : 32 \times C \times H \times W$ for input feature with 32 temporal length. "$\otimes$" denotes matrix Hadamard product, and "$\oplus$" denotes element-wise sum. Here, input dimension is reduced to $d = 128$.

branch in MGTE block, EAC-Net keeps initial behavior of its C2D counterpart at start of training, but finally it would learn more powerful video features.

# Approach

## Overall Architecture

We can partition the construction of most action recognition architectures into three main components: 2D CNN backbone for spatial modeling, temporal modeling block and indispensable classifier. For example, C2D leverages the strong spatial representation of 2D CNNs and its temporal module is only a global temporal pooling applied on deep features in the head. C2D is favored by its computational efficiency, but the temporal module is too trivial, resulting in unsatisfied performance on video data. Motivated by above analysis, we proposed a novel temporal block for increasing the capacity of temporal modeling in EAC-Net.

Table 1 shows the overall architecture of our proposed network. EAC-Net is constructed from ResNet (He et al. 2016) architecture. We propose MGTE blocks for temporal modeling which are added right after residual blocks of Res4_1 and Res5_1, i.e. the first block of Res4 and Res5. The resulting feature is then feed into our proposed ATE block to form the final representation vector. Details about the MGTE block and ATE block is given in the following sections.

## MGTE block

As shown in Figure 1 MGTE block is a two-branch architecture, which is composed of a temporal pooling branch and a temporal relation branch. As its name suggests, the temporal relation branch aims to encode temporal relations among neighbor frames complementing to the representation from pooling branch, where gating mechanism is further applied for enhancing the capacity of MGTE block. Specifically, denoting the input tensor as $X_{in}$ with a tensor size of $T \times C \times H \times W$, where H and W for spatial size and T for temporal size. First, we reduce the dimension of $X_{in}$ by $1 \times 1$ spatial convolution followed by a batch norm and ReLU layer, resulting in a tensor $X_d \in R^{T\times d\times H\times W}$, then we split $X_d$ to two pairs by rearranging its temporal orders, resulting in $X_{2t} = X_d[0, 2, 4, ...; :; :; :] \in R^{T/2\times d\times H\times W}$, $X_{2t+1} = X_d[1, 3, 5, ...; :; :; :] \in R^{T/2\times d\times H\times W}$ to facilitate the batch processing of every two neighboring frames. First, $X_{2t}$ and $X_{2t+1}$ are concatenated to form $X_{pairs} \in R^{T/2\times 2d\times H\times W}$ along the axis of feature dimension. Then, the temporal relations can be computed as following:

$$T_{pairs} = relu(bn(k_p * X_{pairs})) \tag{1}$$

Where $k_p \in R^{d\times 2d\times 1\times 1}$ denotes the kernel of the 2D convolution layer, which encodes appearance relations between inter-frames into temporal features $T_{pairs} \in R^{T/2\times d\times H\times W}$.

Inspired by (Feichtenhofer, Pinz, and Wildes 2017; Sun et al. 2018), we further refine the above temporal feature by gate controlled by motion features estimated using neighbor frames. For computational efficiency, the motion feature is approximated by temporal gradient of neighbor features, thus we have:

$$\begin{cases} X_{diff} = |X_{2t} - X_{2t+1}| \\ Gate = \sigma(bn(k_g * X_{diff})) \end{cases} \tag{2}$$

Here, $\sigma$ denoted for sigmoid function, $k_g \in R^{1\times d\times 1\times 1}$ is the kernel of a $1 \times 1$ 2D convolution layer for mapping to gate value space. In practice, we found widening the access to temporal context to control the gate further improved performance, then we have the gate formulated as:

$$\begin{cases} X_{cat} = Concat(X_{diff}, BC(Mean(X_d))) \\ Gate = \sigma(bn(k_g * X_{cat})) \end{cases} \tag{3}$$

Here, $Mean(\cdot)$ denoted the operation for reduction of temporal dimension of a tensor by averaging along that axis. $Concat(\cdot)$ is applied along the 1st axis of input tensor and $BC(\cdot)$ for broadcasting elements of a tensor to its temporal dimension. $k_g \in R^{1\times 2d\times 1\times 1}$is the kernel of a $1 \times 1$ 2D convolution layer for mapping to gate map. Gated temporal features can be formulated as:

$$T_{gate} = T_{pair} \circ Gate \tag{4}$$

The gate is conducted spatially for input tensor, aiming to selectively process information for different spatial subregions. Finally, we formulate the whole MGTE block as:

$$T_{out} = T_{pool} + T_{gate} \tag{5}$$

Here, $T_{pool} \in R^{T/2\times C\times H\times W}$is computed through temporal max pooling $X_{in}$ with kernel size $2\times1\times1$. $k_u \in R^{c\times d\times 1\times 1}$ is for expanding the dimension of $T_{gate}$ to match that of $T_{pool}$.

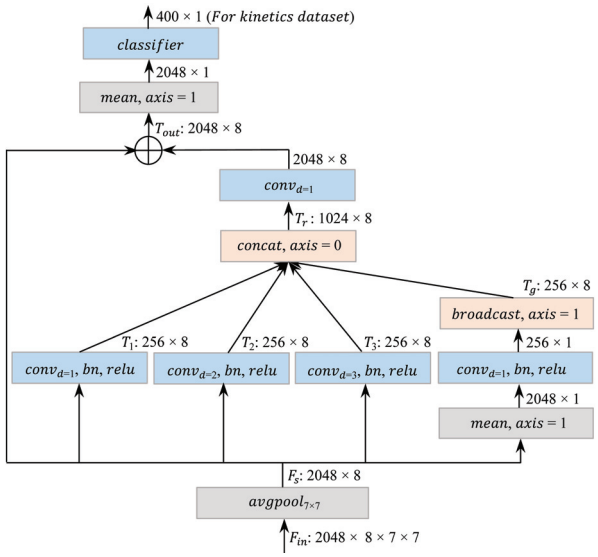


Figure 2: Architecture of ATE block, where the shapes of feature maps follow two types of formats: $C \times T \times H \times W$ for channel, temporal and spatial sizes, and $C \times T$ for channel and temporal sizes, representing features squeezed along spatial dimension. Here, "$\oplus$" denotes element-wise sum. The subscript "d" denotes dilation rate.

## ATE Block

We propose ATE block for further encoding the temporal relations among video snippets at multiple time scales. Specifically, inputs to ATE block are global averaged across spatial dimension, resulting in a tensor $F_s \in R^{C\times T}$, of which each column vector represents a short video snippet. Similarly as done in (Chen et al. 2018a),we then define three 1D convolutions with dilation rates 1, 2 and 3, each kernel tensor is formulated as $k_i \in R^{d\times C\times 3}, i \in [1, 2, 3]$, aiming to capture temporal relations at multiple time scales. Next, $F_s \in R^{C\times T}$ is temporally averaged followed by a temporal convolution $k_4 \in R^{d\times C\times 1}$, leading to holistic temporal context. Thus, we formulate the whole ATE block as:

$$\begin{cases} T_1 = relu(bn(k_1 * F_s)) \\ T_2 = relu(bn(k_2 * F_s)) \\ T_3 = relu(bn(k_3 * F_s)) \\ T_g = BC(relu(bn(k_4 * Mean(F_s)))) \\ T_r = Concat(T_1, T_2, T_3, T_g) \\ T_{out} = F_s + K_u * T_r \end{cases} \tag{6}$$

Here, $Mean(\cdot)$ and $BC(\cdot)$ are applied along $2^{nd}$ axis of input tensor, and $Concat(\cdot)$ applied along the $1^{st}$ axis of input tensor. $k_u \in R^{C\times 4d\times 1}$ is for expanding the dimension of $T_r$ to match that of $F_s$. $T_{out} \in R^{C\times T}$ represents an ensemble of features which encode relations among video snippets at multiple time scales and also holistic temporal context. As shown in Figure 2, $T_{out}$ is averaged along temporal dimension (i.e. the $2^{nd}$ axis of $T_{out}$) to obtain a video level representation, which is then fed into following classifier.

# Experiments

## Datasets

We perform comprehensive studies on the challenging Kinetics dataset (Kay et al. 2017). This dataset contains 246k training videos and 20k validation videos. It is a classification task involving 400 human action categories. We train all models on the training set and test on the validation set.

Other datasets reported are Something-something V1 (Goyal et al. 2017) which consists of 110k videos of 174 different low-level actions. In contrast to Kinetics, this dataset requires making fine-grained low-level distinctions, some of the ambiguous activity categories are challenging, such as 'Turn something upside down' *vs.* 'Pretending to turn something upside down'. We also report results on this dataset to show the generality of our models.

## Implementation Details

**During training.** We first sample 32 frames at random rate from a video, and resize shorter side of each sampled frame to a number prepicked randomly from 215 to 345. Then $224 \times 224$ randomly cropping is applied to these processed frames, leading to the network input with dimension of $32 \times 3 \times 224 \times 224$. In all experiments, our models are initialized by ImageNet (Russakovsky et al. 2015) pre-trained models. For Kinetics, we train for up to 60 epochs, starting with a learning rate of 0.001 and a $10\times$ reduction of learning rate respectively at 30, 50 epoch. We use a momentum of 0.9 and a weight decay of 5e-4. We then fine-tuned models pre-trained on Kinetics to Something-Something V1 dataset, where fine-tuning is conducted for 25 total epochs, starting with initial learning rate 0.001 and reduced by a factor of 0.1 respectively at 10, 15, 20 epoch.

**During inference.** Many state-of-the-art methods use different post-processing techniques during the testing stage. For example, Nonlocal (Wang et al. 2018b) and SlowFast (Feichtenhofer et al. 2019) sample multiple subsets of the video densely and give the averaged output as the prediction, where model inference on each of the subsets is required, resulting in a large amount of computation. Since we target at making the inference efficient and fast, in our methods, 32 frames are first evenly sampled from a video, and then we apply single crop per frame leading to only 32 processed frames for a video to evaluate our models. Specifically, for Kinetics, short sides of sampled frames are rescaled to 256 pixels and then $256 \times 256$ center cropping is applied to the resized frames. Next, we perform fully convolutional inference on these processed frames. For evaluations on Something-Something V1, we just modify the rescaled size and center cropping size to 224 and $224 \times 224$ respectively.

## Ablation Study

We conduct ablation study on Kinetics datasets. All models are equipped with ResNet50 as backbone. Top-1 accuracy (%), as well as computational complexity measured in FLOPs are showed to compare model performance.

**Location and number of MGTE block.** ResNet architecture can be divided into 5 stages. We refer the Res3_x to

| Model, R50 | Res3_1 | Res4_1 | Res5_1 | Top-1 | FLOPs |
|---|---|---|---|---|---|
| C2D | \ | \ | \ | 71.0 | 172.7G |
| EAC-Net w/o ATE BLock | ✓ | × | × | 72.6 | 112.9G |
| | × | ✓ | × | 73.6 | 134.5G |
| | × | × | ✓ | 73.7 | 165.3G |
| | × | ✓ | ✓ | 74.3 | 130.8G |
| | ✓ | ✓ | ✓ | 73.2 | 91.9G |

Table 2: Results for adding MGTE blocks to different stages of backbone.

| Model, R50 | Temporal Pool | Temporal Relation | Top-1 | FLOPs |
|---|---|---|---|---|
| C2D | × | × | 71.0 | 172.7G |
| EAC-Net w/o ATE BLock | ✓ | × | 72.0 | 128.2G |
| | ✓ | ✓ | 74.3 | 130.8G |

Table 3: The effect of temporal pooling branch and temporal relation branch in MGTE block. Here, MGTE blocks are applied on to stage 4 and stage 5.

Res5_x as stage 3 to stage 5. The $2^{nd}$ to $4^{th}$ rows of Table 2 compare the performance of adding single MGTE block right after only the first residual block on different stages in ResNet-50, from stage 3 to stage 5, respectively. We conclude from the results that adding only one MGTE block already yield significant performance improvement compared to the baseline C2D in Table 1, which demonstrates the effectiveness of the proposed MGTE block. Besides, adding the MGTE block at latter stage (e.g., stage 5) yield better accuracy than early stage (e.g., stage 3). One possible reason is that temporal modeling is beneficial more with larger receptive fields that can capture global temporal features. We then add multiple MGTE blocks to ResNet-50 and leads to the best result when adding two MGTE blocks respectively to the stage 4 and stage 5. Next, we further add a MGTE block to stage 3 and find from the results that the model suffers top-1 accuracy degradation of 1.1% (73.2% *vs.* 74.3%), which may be due to the fact that too early spatial temporal fusion does harm to low-level feature learning (Carreira and Zisserman 2017).

**Ablation study on branches in MGTE block.** In Table 3, with respect to the C2D baseline with no temporal modeling at all, adding MGTE blocks with only pooling branch boosts the model accuracy by 1% and significantly reduces the FLOPs from 172.7G to 128.2G due to the temporal down sampling by pooling branch. For fitting our earlier analysis in paper, we also add the temporal relation branch for complementing to the pooling branch. We can see that the accuracy is boosted by a large margin, 2.3% increased, while the FLOPs are only increased from 128.2G to 130.8G.

**Impact of gate on temporal relation representation.** In Table 4, we evaluated some variants of EAC-Nets on Kinetics, which differ in their respective implementations of MGTE blocks. The first two rows in Table 4 compare the results of two variant models: one is equipped with modified MGTE blocks whose gate mechanism is entirely disabled, and the other a variant model with gate operation enabled and controlled by temporal gradient of consecutive features. From the results, we found that introducing the gate into the temporal relation branch brings 0.6% top-1 accuracy improvement, demonstrating the effectiveness of the gate controlled by approximated motion features. Furthermore, applying holistic temporal context as an additional component of controlling signals, can further promote model performance. It is given in the $3^{rd}$ line of Table 4 that 0.7% top-1 improvement is gained from adding holistic temporal context as a control component.

| Model, R50 | Temporal Gradient | Global Context | Top-1 |
|---|---|---|---|
| EAC-Net w/o ATE BLock | × | × | 73.0 |
| | ✓ | × | 73.6 |
| | ✓ | ✓ | 74.3 |

Table 4: Ablation study on the gate in MGTE block. Results from the first two rows correspond to model configs using MGTE blocks w. or w/o. enabling gate unit respectively. Each component for controlling the gate is enabled one by one, and the progressively improved performance proves their individual effectiveness.

| Model, R50 | MGTE Block | ATE Block | Top-1 | FLOPs |
|---|---|---|---|---|
| C2D | × | × | 71.0 | 172.7G |
| | × | ✓ | 72.5 | 173.6G |
| EAC-Net | ✓ | × | 74.3 | 130.8G |
| | ✓ | ✓ | 74.8 | 131.1G |

Table 5: Ablation study on ATE block on Kinetics. Noting that MGTE blocks are applied on to stage 4 and stage 5 in EAC-Net.

**Impact of ATE block.** Here, we choose two baselines to prove the effectiveness of ATE block; one is the best-performing model in Table 2 where MGTE blocks are applied in stage 4 and stage 5; another is the C2D counterpart described in Table 1. No ATE block is employed in both of the baseline models. By observing the results in Table 5, it would produce better accuracy w.r.t both of the baselines that modifying the global average pooling in the head to ATE block: top-1 accuracy of the C2D is further boosted to 72.5% (71.0% *vs.* 72.5%), and for the other model, incorporating ATE block brings 0.5% accuracy improvement. Moreover, the inference FLOPs of both models are slightly increased by 0.9G and 0.3G respectively. It evidences the fact that it is necessary for enhancing performance of a video model that modeling temporal interactions among the feature sequences at different temporal scales.

## Comparison with Other Methods on Kinetics

We evaluate our proposed framework against the recent state-of-the-art methods on Kinetics-400 in terms of their effectiveness (i.e. top-1 accuracy) and efficiency measured by number of FLOPs. Each of the approaches is instantiated using different 2D CNN backbones. Due to these methods differ in their inference strategy for cropping/clipping in space and in time, each method is evaluated using different numbers of temporal clips and spatial crops as input during inference, and their top-1 accuracy and FLOPs are reported in Table 6. For better analysis of accuracy-cost trade-off, we apply the chart in Figure 3 to visualize the results in Table 6, where the points in different shapes marked in Figure 3 are

| Methods | Backbone | Frame×Clip | Input Size | Top-1 | FLOPs |
|---|---|---|---|---|---|
| I3D | InceptionV1 | 319* × 1 | 256 × 256 | 70.2 | 544G |
| | R50 | 32 × 30 | 256 × 256 | 73.3 | 1260G |
| | R101 | 32 × 30 | 256 × 256 | 74.4 | 2010G |
| Nonlocal-I3D | R50 | 32 × 30 | 256 × 256 | 74.9 | 1470G |
| | R101 | 32 × 30 | 256 × 256 | 76.0 | 2190G |
| CoST | R50 | 8 × 10 | 256 × 410* | 74.1 | 750G |
| | R101 | 8 × 10 | 256 × 410* | 75.5 | 1370G |
| Slowfast(4 × 16) | R50 | 32 × 30 | 256 × 256 | 75.6 | 1080G |
| Slowfast(8 × 8) | R50 | 64 × 30 | 256 × 256 | 77.0 | 1980G |
| Slowfast(4 × 16) | R101 | 32 × 30 | 256 × 256 | 76.9 | 1740G |
| MF-Net | Custom | 16 × 1 | 224 × 224 | 65.0 | 11G |
| | Custom | 16 × 50 | 224 × 224 | 72.8 | 555G |
| StNet | R50 | 125 × 1 | 256 × 256 | 69.9 | 189G |
| | R101 | 125 × 1 | 256 × 256 | 71.4 | 310G |
| TSM | R50 | 8 × 1 | 224 × 224 | 71.2 | 33G |
| | R50 | 16 × 1 | 224 × 224 | 72.6 | 65G |
| | R50 | 8 × 10 | 224 × 224 | 72.8 | 330G |
| | R50 | 16 × 10 | 224 × 224 | 73.7 | 650G |
| | R50 | 8 × 30 | 256 × 256 | 74.1 | 1290G |
| **EAC-Net** | **BNInception** | **32 × 1** | **256 × 256** | **73.5** | **61G** |
| | **BNInception** | **48 × 1** | **256 × 256** | **74.2** | **91G** |
| | **R50** | **32 × 1** | **256 × 256** | **74.8** | **131G** |
| | **R50** | **48 × 1** | **256 × 256** | **75.3** | **196G** |
| | **R101** | **32 × 1** | **256 × 256** | **75.5** | **209G** |
| | **R101** | **48 × 1** | **256 × 256** | **76.3** | **313G** |
| | **R152** | **32 × 1** | **256 × 256** | **76.6** | **305G** |
| | **R152** | **48 × 1** | **256 × 256** | **77.2** | **458G** |

Table 6: Ablations on Kinetics-400 action recognition. We show Top-1 classification accuracy (%), as well as computational cost measured in FLOPs. Numbers with " *" are taken as the averages of varying sizes of data fed to the model when scanning all over the validation set.

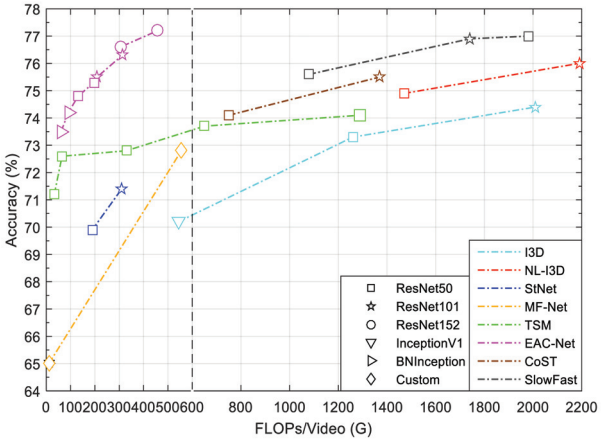


Figure 3: Comparisons of EAC-Net with previous approaches for video classification. Models from these approaches are instantiated with different backbones and evaluated using different test strategies. Our EAC-Net significantly outperform other competitors by better trade-off between accuracy and cost, and provide a new upper envelope in the accuracy-cost plot.

associated with all the lines of results in Table 6 in a one-to-one correspondence. Besides, the points corresponding to results from the same family of models are connected with dashed lines in the same color, aiming to make it possible to intuitively learn about the sensitivity of model accuracy to inference FLOPs. We mainly compare our framework with two categories of approaches, where one category of methods focuses on designing efficient models for achieving better trade-off between accuracy and cost, the other includes high-performance approaches with less regards to computation complexity at inference. Noting that models reported in Figure 3 are all under 2200G FLOPs at inference, considering that models with more than 2200G FLOPs thus can't be efficiently used in application.

**For category of cost-efficient approaches.** Several efficiency-oriented approaches such as MF-Net (Chen et al. 2018b), StNet (He et al. 2019), InceptionV1-I3D (Carreira and Zisserman 2017) and TSM (Lin, Gan, and Han 2019) are evaluated and listed in Table 6 for the first category of methods. Compared to TSM, our method always can achieve better performance at smaller FLOPs. Specifically, TSM achieves top-1 accuracy 72.6% with 65G FLOPs and accuracy 72.8% with explosion of FLOPs to 330G, however, our BN-Inception based EAC-Net can beat both the results by achieving top-1 accuracy 73.5% at smaller cost of 61G FLOPs. If increasing the FLOPs of TSM to 650G by enlarging input size to ten 16-frame clips, its accuracy is boosted to 73.7%, while only 0.2% higher than accuracy of our EAC-Net (73.7% *vs.* 73.5%), but at huge cost of about 10× more FLOPs than that of EAC-Net (650G *vs.* 61G). By lengthening the input clip from 32 to 48 frames, the performance of EAC-Net can be further boosted to 74.2% at a cost of only 91G FLOPs, surpassing the result of TSM over both accuracy and efficiency. Compared with the heaviest model from TSM family, which applies 30 8-frame clips as input and achieves top-1 of 74.1% with 1290G FLOPs, BN-Inception based EAC-Net can still present higher accuracy than TSM with over about 14× FLOPs reduction. MF-Net is very cheap in FLOPs (11.1G FLOPs), but its performance is poor (65.0%) if testing 1 clip. When 50 clips are used, top-1 is boosted to 72.8% at an increased cost of 555G. In a similar case, the model size and performance of InceptionV1-I3D is plausible, however, applying itself convolutional over whole frames at inference leads to a cost of more than 500G FLOPs. Our BN-Inception based EAC-Net outperform these models by a large margin over accuracy and FLOPs, showing a better trade-off.

**For category of high-performance approaches.** We first perform apple-to-apple comparisons between EAC-Net and Non-Local I3D by using same 2D backbones. As shown in Table 6, EAC-Net models with 48-frame clip as input can achieve 0.3% better accuracy with about 7× fewer FLOPs compared to Non-local I3D counterparts which take more than 1000G FLOPs at inference. SlowFast (Feichtenhofer et al. 2019) network is a most recent state-of-the-art method, and it can achieve 77% top-1 accuracy with 1980G FLOPs. Our heaviest EAC-Net can achieve 77.2% accuracy with

| Methods | Backbone | #Frame×Clip | Top-1 | FLOPs |
|---|---|---|---|---|
| C2D | ResNet50 | $32 \times 1$ | 22.8 | 132G |
| TRN-Multiscale | BNInception | $8 \times 1$ | 34.4 | 16G |
| TRN-Multiscale | BNInception | $8 \times 1$ | 38.9 | 33G |
| ECO | BNInception+R18 | $8 \times 1$ | 39.6 | 32G |
| ECO | BNInception+R18 | $16 \times 1$ | 41.4 | 64G |
| ECO$_{en}$Lite | BNInception+R18 | $92 \times 1$ | 46.4 | 267G |
| I3D | ResNet50 | $32 \times 2$ | 41.6 | 306G |
| Nonlocal-I3D | ResNet50 | $32 \times 2$ | 44.4 | 336G |
| Nonlocal-I3D+GCN | ResNet50+GCN | $32 \times 2$ | 46.1 | 606G |
| TSM | ResNet50 | $8 \times 1$ | 43.4 | 33G |
| TSM | ResNet50 | $16 \times 1$ | 44.8 | 65G |
| TSM$_{en}$ | ResNet50 | $24 \times 1$ | 46.8 | 98G |
| **EAC-Net** | **BNInception** | **$32 \times 1$** | **44.5** | **47G** |
| | **BNInception** | **$48 \times 1$** | **47.0** | **70G** |
| | **ResNet50** | **$32 \times 1$** | **45.6** | **100G** |
| | **ResNet50** | **$48 \times 1$** | **47.4** | **150G** |

Table 7: Comparisons of EAC-Net with state-of-the-art approaches on Something-Something V1 dataset.

only 458G FLOPs, about $4.3\times$ fewer FLOPs than Slow-Fast net. To the best of our knowledge, EAC-Net is the first method whose top-1 accuracy could reach above 77% on Kinetics validation sets but with a cost lower than 500G FLOPs. By observing the chart in Figure 3, models from EAC-Net family cluster in the upper left region of the chart, matching high precision and low computational cost performance.

### Transfer Learning on Something-Something V1

We transfer the models of EAC-Net pre-trained on Kinetics to Something-Something V1 dataset to show the learned representation can be well generalized to video dataset which heavily relies on temporal relationships. In Table 7 we present the TRN (Zhou et al. 2018a) models, where only late temporal fusion is added after feature extraction, leading to results significantly lower than state-of-art methods. Then compared with ECO (Zolfaghari, Singh, and Brox 2018), our method achieves better performance at a smaller FLOPs. For instance, our BN-Inception model achieves 44.5% with 47G FLOPs, which is 3.1% more accurate than ECO while taking about 27% less computational cost (47G *vs.* 64G). The ensemble version of ECO achieves top-1 accuracy 46.4% with the FLOPs increased to 267G, However, our BN-Inception based EAC-Net still presents better performance by 0.6% more accuracy and $3.8\times$ less FLOPs than that of ensemble ECO (70G *vs.* 267G). For comparing with high-performance methods, our BN-Inception model can achieve only 0.1% accuracy improvement but with $7\times$ fewer FLOPs compared to the Non-local I3D network (47G *vs.* 336G). Although the Non-local I3D + GCN (Wang and Gupta 2018) network further boosts the top-1 accuracy to 46.1% with FLOPs as huge as 606G, our EAC-Net still outperforms it by top-1 accuracy of 47% and $8.6\times$ fewer FLOPs (70G *vs.* 606G). Interestingly, we found that there is a large accuracy gap between the results of C2D and our EAC-Net (22.8% *vs.* 44.5%) on Something-Something V1 dataset. A similar case already happened when evaluating TSN (Wang et al. 2016) models on Something-Something V1 dataset, where TSN models achieved bad performance due to the lack of temporal modeling (Lin, Gan, and Han 2019). Therefore, from the result of C2D, we conclude that temporal dimension cannot be trivially addressed by only pooling. In Table 7, the 8, 16-frame models of TSM already present property of cost-efficiency at inference, while they failed to achieve decent accuracy. Ensemble of the 8, 16-frame models promotes the performance to top-1 46.8% but at the cost increased to 98G FLOPs, however, which is still 0.2% less accurate and $1.4\times$ more costly than our BN-Inception model (98G *vs.* 70G).

## Conclusion

In this paper, we have proposed the EAC-Net for efficient and accurate video recognition, which can be implemented using common pre-trained 2D CNNs as backbone, and incorporating our proposed MGTE blocks for local temporal modeling and ATE block for late temporal fusion to empower the model to learn video-level representation. Experiments on large-scale benchmark Kinetics have verified the effectiveness and cost-efficiency of EAC-Net. In addition, EAC-Net trained on Kinetics also exhibits pretty good transfer learning ability on the Something-Something V1 dataset.

## Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments, and also would like to thank all their friends, especially Xu's neighbour, for her encouragement and support.

## References

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

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018a. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; and Feng, J. 2018b. Multi-fiber networks for video recognition. In *The European Conference on Computer Vision (ECCV)*.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *The IEEE International Conference on Computer Vision (ICCV)*.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2017. Spatiotemporal multiplier networks for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Goyal, R.; Ebrahimi Kahou, S.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; Hoppe, F.; Thurau, C.; Bax, I.; and Memisevic, R. 2017. The "something something" video database for learning and evaluating visual common sense. In *The IEEE International Conference on Computer Vision (ICCV)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 770–778.

He, D.; Zhou, Z.; Gan, C.; Li, F.; Liu, X.; Li, Y.; Wang, L.; and Wen, S. 2019. Stnet: Local and global spatial-temporal modeling for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 8401–8408.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence (ICML)* 35(1):221–231.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Laptev, I.; Marszalek, M.; Schmid, C.; and Rozenfeld, B. 2008. Learning realistic human actions from movies. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Laptev, I. 2005. On space-time interest points. *International journal of computer vision (IJCV)* 64(2-3):107–123.

Li, Y.; Qi, H.; Dai, J.; Ji, X.; and Wei, Y. 2017. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2359–2367.

Li, C.; Zhong, Q.; Xie, D.; and Pu, S. 2019. Collaborative spatiotemporal feature learning for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7872–7881.

Lin, J.; Gan, C.; and Han, S. 2019. Tsm: Temporal shift module for efficient video understanding. In *The IEEE International Conference on Computer Vision (ICCV)*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision (ECCV)*, 21–37. Springer.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 3431–3440.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 5533–5541.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NIPS)*, 91–99.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)* 115(3):211–252.

Scovanner, P.; Ali, S.; and Shah, M. 2007. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia (ACM MM)*, 357–360. ACM.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems (NIPS)*, 568–576.

Sun, L.; Jia, K.; Yeung, D.-Y.; and Shi, B. E. 2015. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 4597–4605.

Sun, S.; Kuang, Z.; Sheng, L.; Ouyang, W.; and Zhang, W. 2018. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 1390–1399.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 4489–4497.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 6450–6459.

Wang, X., and Gupta, A. 2018. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 399–417.

Wang, H., and Schmid, C. 2013. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 3551–3558.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision (ECCV)*, 20–36. Springer.

Wang, L.; Li, W.; Li, W.; and Van Gool, L. 2018a. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 1430–1439.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018b. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7794–7803.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 305–321.

Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; and Agrawal, A. 2018. Context encoding for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7151–7160.

Zhou, B.; Andonian, A.; Oliva, A.; and Torralba, A. 2018a. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 803–818.

Zhou, Y.; Sun, X.; Zha, Z.-J.; and Zeng, W. 2018b. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 449–458.

Zolfaghari, M.; Singh, K.; and Brox, T. 2018. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 695–712.