

Stacked Temporal Attention: Improving First-person Action Recognition by Emphasizing Discriminative Clips

Lijin Yang
yang-lj@iis.u-tokyo.ac.jp

Yifei Huang
hyf@iis.u-tokyo.ac.jp

Yusuke Sugano
sugano@iis.u-tokyo.ac.jp

Yoichi Sato
ysato@iis.u-tokyo.ac.jp

Institute of Industrial Science
The University of Tokyo
Tokyo, Japan

Abstract

First-person action recognition is a challenging task in video understanding. Because of strong ego-motion and a limited field of view, many backgrounds or noisy frames in a first-person video can distract an action recognition model during its learning process. To encode more discriminative features, the model needs to have the ability to focus on the most relevant part of the video for action recognition. Previous works explored to address this problem by applying temporal attention but failed to consider the global context of the full video, which is critical for determining the relatively significant parts. In this work, we propose a simple yet effective Stacked Temporal Attention Module (STAM) to compute temporal attention based on the global knowledge across clips for emphasizing the most discriminative features. We achieve this by stacking multiple self-attention layers. Instead of naive stacking, which is experimentally proven to be ineffective, we carefully design the input to each self-attention layer so that both local and global context of the video is considered during generating the temporal attention weights. Experiments demonstrate that our proposed STAM can be built on top of most existing backbones and boost the performance in various datasets.

1 Introduction

Video action recognition is one of the most fundamental problems in computer vision. Most research focus on recognizing human actions from videos taken by fixed cameras [1, 2, 3] (third-person videos). Different from these third-person videos, first-person videos taken by wearable cameras record human behaviors from a different perspective, with a wide range of applications such as assisting human-computer interaction. Unfortunately, the performance of action recognition methods in first-person videos are still not comparable to that in third-person videos [4].



Figure 1: Example of action “turn off faucet” in the EGTEA dataset [25]. Only the frames with green boundaries provide enough information for determining the action. In all other frames, the action happens outside the visible field-of-view.

One of the major challenges for first-person action recognition is that the unique field of view makes actions sometimes happen outside the video viewing range (Figure 1). Since the action is not always observable in all frames of the video sequence, the methods that first sparsely sample clips from the video and then compute the clip consensus via pooling [23, 42], which are proved to be good practices for creating more discriminative features in third-person videos, cannot achieve as good performance in first-person videos.

Another major challenge for first-person action recognition is that the videos are often accompanied by huge ego-motion caused by the sharp movements of the camera wearer (e.g. turning head), which complicates the encoding of motion [40]. With current clip sampling techniques, some of the clips may contain purely background information that harms the recognition, even with an enlarged temporal receptive field.

One way to address the two challenges jointly is to apply temporal attention to the clips, emphasizing the discriminative clips while suppressing the noisy ones. Several previous works have explored this direction. For example, Pei *et al.* [35] propose to use gated LSTM for deciding the weight of each input frame, and Girdhar *et al.* [9] leverage transformers [42] to compute spatio-temporal attention on humans in the video. However, either using LSTM or transformers, the temporal information is only formulated by computing clip-wise relationships. It is still difficult for the model to decide the best choice of attention without a global understanding of the whole input. Intuitively thinking, global knowledge of all the local clips can help to identify which clip is the most discriminative one and suppress the less important clip features.

In this work, building upon this intuition we propose a Stacked Temporal Attention Module (STAM) that leverages both local information from individual clips and global information from the whole video to generate temporal attention. We find that this could be done by simply stacking self-attention [42] layers with global information as the query vectors. It is worthwhile to note that our STAM can be built on top of most existing action recognition backbones. Experiments on multiple first-person datasets demonstrate that by adding STAM on top, the action recognition performance could be improved. Further studies on HMDB51 dataset [21] show that our method can even generalize to third-person datasets.

Our major contributions are: (1) We propose a simple yet effective Stacked Temporal Attention Module (STAM) that could be built on top of most existing backbones for improving first-person action recognition. (2) Experiments on multiple datasets demonstrate that our proposed module can improve the performance of various backbone models.

2 Related Work

First-person action recognition Action recognition is one of the most focused research areas in computer vision, and first-person action recognition is also extensively studied. Earlier works design different hand-crafted features [1, 24] to represent the rich spatio-temporal information in the video sequence. With the development of deep learning, remarkable progress has been made recently [8, 15, 16, 17, 23, 29, 30, 39]. Since temporal information is essential for recognizing the actions in the videos, a line of research [27, 40] uses recurrent networks for sequential modeling. However, one major drawback of using recurrent models is that the span of attention is limited [37], which prevents these models to perform well when the input video is long. Some previous works also tried to incorporate unique first-person cues such as hand and gaze [13, 14, 29, 38, 50], or use multiple modalities [18] for improving first-person action recognition. Some researchers try to input multiple frames and use spatio-temporal convolution networks for aggregating features both spatially and temporally [25, 45]. One problem of 3D convolution networks is that the limited temporal receptive field results in a model that lacks long-term temporal knowledge of the input. In this work, we use these models as backbones and build our Stacked Temporal Attention Module on top for further improving their performances.

Temporal attention for action recognition Previous works tried different temporal attention methods to aggregate the temporal features for longer-term video understanding and have shown promising results in action recognition of third-person videos. One line of research use temporal aggregation methods to implicitly apply different weights on each feature [11, 47, 49]. For example, TSN [49] sparsely samples frames from videos and assign uniform temporal attention by average pooling to combine features of each frame. TLE [9] designed a temporal linear encoding layer for generating weight for each temporal input.

Another line of research explicitly computes a temporal attention score for each feature and uses a weighted average for the aggregation. As for previous works, TAGM [35] modified the LSTM block and sequentially determine a weight for each input frame. One major drawback is that LSTM cannot access global information, thus the attention produced by only local features is sub-optimal. Similar to our method, CatNet [43] also uses global information to aggregate the clip features. However, they simply concatenate the local and global features and use several fully connected layers for predicting the weights. The attention produced by these fully connected layers is used as temporal attention weights.

In this work, we design a simple yet effective module that could be built on top of most action recognition models for computing temporal attention. We find that simply by stacking self-attention layers [42] with global feature vectors as queries, we can improve recognition performance on multiple first-person datasets.

3 Approach

In this section, we describe our proposed Stacked Temporal Attention Module (STAM). Different from existing methods for temporal attention [35, 44, 49], we consider the global information across clips, and use the relation between a clip and the global representation to determine the significance of each clip. An overview of our model architecture is depicted in Figure 2.

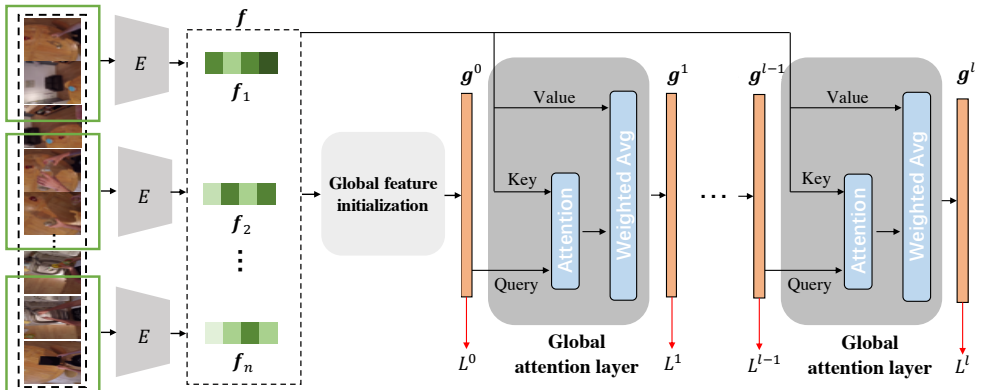


Figure 2: Overview of our proposed STAM. Using the features from the backbone, a global feature \mathbf{g}^0 is first initialized. This global feature serves as an aggregated global information of all input clips and is used for the following global attention layers. The stack of global attention layers progressively refines the attention weight as well as the global feature. Classification loss is applied to the output of each layer.

Backbone encoder Our STAM can cooperate with most of the existing backbones that use a neural network and back-propagation for training. Here we showcase a backbone encoder E that uses 3D convolution and takes as input a small video clip with a few video frames. Given a trimmed video containing an action, we follow the practice of TSN [14] to split the input into N segments. From each segment we extract a clip of 16 consecutive frames as inputs to the backbone network. The backbone would encode the clips into clip features $\mathbf{f}_1, \dots, \mathbf{f}_N$. These clip features are used as inputs to our STAM.

3.1 Stacked Temporal Attention Module

With the features encoded by the backbone, our STAM finds attention for temporally aggregating the features using both local and global information. As shown in Figure 2, the STAM is a stack of several global attention layers. The input to the first global attention layer is the global feature vector \mathbf{g}^0 initialized by the local feature vectors \mathbf{f} . We explore multiple initialization methods in this work. In the l -th global attention layer, an attention weight vector \mathbf{a}^l is computed using the relation between the input global feature \mathbf{g}^{l-1} and each local feature \mathbf{f} . This attention weight is used to generate a new global vector \mathbf{g}^l as the output of this layer. For simplicity, we omit all the classifiers and normalization layers in this section.

3.1.1 Global Attention Layer

As in Figure 2, the global attention layer takes the encoded clip features and the global feature \mathbf{g} as inputs. Intuitively, knowing the global context can help to determine the importance of each local clip. Thus, the relationship between the global feature and each of the clip features should be considered to emphasize discriminative clips for getting better temporal attention. This could be neatly done taking advantage of the self-attention operation of the Transformers [17]. Take the first global attention layer as an example, the query vector is acquired using the global feature \mathbf{g}_0 , while the key and value vectors are from each local features \mathbf{f} :

$$\mathbf{q}_i = W_q \mathbf{g}^0, \quad \mathbf{k}_i = W_k \mathbf{f}_i, \quad \mathbf{v}_i = W_v \mathbf{f}_i; \quad i \in [1, N] \quad (1)$$

where W_q, W_k, W_v are the weight matrices for projecting the features to query \mathbf{q} , key \mathbf{k} and value \mathbf{v} , respectively. The queries and keys are vectors with hidden dimension d , and the value vectors have the same dimension as the backbone features. We then compute a weight vector a_i^1 for the i -th clip using a modified scaled dot-product operation and softmax operation:

$$c_i = \frac{1}{N\sqrt{d}} \sum_{j=1}^N \mathbf{q}_i \cdot \mathbf{k}_j, \quad a_i^1 = \frac{e^{c_i}}{\sum_j e^{c_j}} \quad (2)$$

The generated weight vectors are used as temporal attention weights:

$$\mathbf{g}^1 = \sum_{i=1}^N a_i^1 \mathbf{v}_i \quad (3)$$

\mathbf{g}^1 is the output of the first global attention layer and will also be the input to the next layer. We add a classifier on top of \mathbf{g}^1 to recognize actions and omit it in the equations for simplicity.

Stacking several modules sequentially has shown significant improvements in many tasks like human pose estimation [43] and action segmentation [5]. By composing several models sequentially, each model can perform an incremental refinement to the previous output. Inspired by the previous success, in this work we propose to stack multiple global attention layers for better refining the temporal attention and the global features.

Denote the operation of the first global attention layer as $\mathcal{G}^1(\mathbf{g}^0, \mathbf{f}_1, \dots, \mathbf{f}_N)$, the k -th global attention layer can be represented accordingly as follows:

$$\mathbf{g}^k = \mathcal{G}^k(\mathbf{g}_{k-1}, \mathbf{f}_1, \dots, \mathbf{f}_N) \quad (4)$$

We show in our experiments that as the global feature refines, the attention scores also change, giving more focus on the most critical clip from a global perspective.

3.1.2 Global feature initialization

Given the encoded clip features $\mathbf{f}_1, \dots, \mathbf{f}_N$ as inputs, the global feature vector \mathbf{g}^0 can be initialized in multiple ways. We experiment with the following choices while others are possible:

Pooling: One of the most straightforward ways is to do pooling on all clip features to get the global feature: $\mathbf{g}^0 = \text{pool}(\mathbf{f}_1, \dots, \mathbf{f}_N)$. We experiment with both max-pooling and average-pooling in the experiment section.

Recurrent networks: It is also feasible to use recurrent neural network to process all clip features, and generate a global aggregated feature. We choose bi-directional GRU as an example: $\mathbf{g}^0 = \text{GRU}(\mathbf{f}_1, \dots, \mathbf{f}_N)$.

Temporal convolution: Apply a 1D temporal convolution layer to fuse all the clip features: $\mathbf{g}^0 = \text{Conv}(\mathbf{f}_1, \dots, \mathbf{f}_N)$.

Light-weight CNN: Inspired by previous works [20, 48], we also tried to use a light-weight CNN for extracting the global feature. The lightweight CNN is a small 2D CNN [12] that takes one frame from each input clip as inputs. The output features of all inputs are averaged to form the global feature.

Self-attention: We use the self-attention operation of Transformers [42] to compute the pairwise relationship between input clips, and use generated weight to form the global feature.

3.2 Model training

To incrementally refine global features of each layer, we add a classifier with cross-entropy loss on each attention layer during training the model. The final loss function is a combination of the loss from both the global feature initialization step and M global attention layers:

$$L = \sum_{i=0}^M \lambda_i L^i \quad (5)$$

Here $\lambda_0, \dots, \lambda_M$ is a set of model hyper-parameters to determine the contribution of the different losses.

4 Experiments and Results

We conduct experiments on two publicly available first-person datasets: EGTEA [25, 26] and EPIC-Kitchens-100 [3] (EPIC-100). EGTEA is an egocentric video dataset containing 29 hours of meal preparing actions performed in a kitchen by 32 subjects. Fine-grained annotations of 106 action classes are provided. Following [40], we report the average of three splits of the dataset. EPIC-100 is the currently largest dataset of first-person videos, with 100 hours of videos recorded at different kitchens by different subjects. Fine-grained annotations of 97 verbs and 300 nouns are provided. We follow the official protocol and report the performance of verb, noun and the combined actions. For all datasets, we report the action recognition accuracy as the evaluation metric.

Implementation Details. We use PyTorch [52] for all the implementation. All backbones are pretrained on Kinetics dataset [10]. We train STAM together with pretrained backbone in the end-to-end manner using the Adam optimizer [19] with initial learning rate $1e-4$ for 40 epochs. As for the loss weight λ we empirically set all the λ s to 1.

The hidden dim d is set as 512 in all experiments. For the 2D backbone TSM, we uniformly sample 16 frames as the input. The output temporal dimension is the same as clip number for both 2D and 3D backbones. For the number of global attention layer M , we empirically set different values to cooperate with different backbone encoders. More details about implementation can be found in the supplementary material.

4.1 Initialization options of global feature

One core component in our STAM is the use of global feature \mathbf{g} for temporal attention computation. It is possible to utilize multiple alternatives as the initialization of global feature as described in Section 3. Table 1 shows the performance of our STAM when changed with different alternatives of global feature \mathbf{g} . We further add directly stacking self-attention layers as the vanilla stacking baseline. Note that vanilla stacking baseline does not include global feature initialization, we use one more self-attention layer instead to keep layer number fair with our STAM using self-att initialization. From Table 1, we can see that our proposed STAM is not too sensitive about the initialization option of global feature. We empirically find that using self-attention for global feature initialization can get the most stable performance in different settings (*e.g.* different number of input clips). We think one possible reason is that self-attention alone can be a strong baseline for action recognition (validated in Table 2). Thus, we use this option in all of the following experiments unless otherwise stated.

Global feature options	Num of global att layers		
	1	2	3
	Vanilla stacking	65.6	65.5
Max pooling	65.8	65.9	66.3
Avg pooling	65.9	66.3	66.7
Bi-GRU	65.8	66.3	65.9
1D-conv	65.9	66.1	65.9
Self-att	66.2	66.5	66.9
Light-weight CNN	64.7	66.0	67.1

Table 1: Different design options for the global feature. Experiments are conducted on the EGTEA dataset split1 using I3D as backbone and 9 clips as inputs. Dark row indicates no global feature is used.

Method	EGTEA	EPIC-100		
		Verb	Noun	Action
Max pooling	63.6	61.8	45.6	33.7
Avg pooling	63.0	63.2	48.0	36.2
Bi-GRU	62.5	62.3	45.1	33.7
1D-conv	62.6	63.4	46.3	35.2
Self-att	65.5	62.9	48.6	36.8

TAGM [35]	65.4	63.6	45.7	34.3
CatNet [43]	63.2	61.5	48.7	35.8
TLE [9]	63.4	63.6	45.4	34.3
TRN [49]	64.0	64.6	47.9	36.4
MTRN [49]	64.1	64.8	47.6	36.6
GSTA [40]	63.8	62.2	49.1	36.5
STA [27]	63.8	62.6	48.6	36.6
CTA [46]	62.6	62.3	47.2	35.4

Our STAM	66.9	64.4	50.3	38.0

Table 2: Comparison of different feature aggregation methods on EGTEA dataset split1 and EPIC-100 dataset.

It is worthwhile to note that directly stacking self-attention layers can hardly improve model performance, which shows the same trend as in [52]. However, our STAM with all initialization options can benefit from stacking more than one layer. We believe this is because although the vanilla stacking baseline uses updated local feature as the key and query for the next layer, it still cannot access global information of all other local features at the same time. Also, simultaneously updating the key and query can lead to accumulated errors, which may even make performance worse when increasing layers.

Compared with vanilla stacking, our proposed method leverages the global feature and only refines the query when stacking, which enables the model to better determine the relatively significant clips as well as benefit from stacking. Therefore, our STAM (self-attention for global feature initialization) can clearly surpass vanilla stacking baseline under fair comparison. This suggests the usefulness of STAM for generating temporal attention, and shows potential applications in other tasks that also deal with temporal information, *e.g.*, action detection.

4.2 Comparison of temporal attention methods

For evaluation of our proposed temporal attention module, we compare our STAM with some baselines and several state-of-the-art temporal attention methods. For EGTEA dataset we utilize I3D backbone and use 9 clips with 16 frames for all methods. For EPIC-100 dataset we use TSM backbone with 16 frames as inputs. We specifically compare with the following clip feature aggregation methods:

- Global feature initialization baselines without the proposed STAM. These baselines serve as simple straightforward ways for temporal aggregation.
- TLE [9], TAGM [35], CatNet [43], TRN [49], GSTA [40], STA [27] and CTA [46] are previous works that uses various algorithms for temporal aggregation.
- Our STAM. In this experiment, for EGTEA and EPIC-100 we use STAM with three global attention layers and two global attention layers respectively. Studies on the influence of the number of global attention layers are placed in Section 4.4.

Method	Split1	Split2	Split3	Average
Ego-RNN [19]	62.2	61.5	58.6	60.8
LSTA* [10]	-	-	-	61.9
SAP [15]	64.1	62.1	62.0	62.7
Lu <i>et al.</i> [25]	63.7	61.1	59.0	61.3
Min* <i>et al.</i> [5]	69.6	-	-	-
TSM [28]	63.8	61.8	60.2	61.9
I3D [9]	63.0	61.1	58.0	60.7
R3D-50 [10]	63.2	61.4	59.4	61.3

TSM + STAM	66.2	64.1	64.0	64.8
I3D + STAM	66.9	63.8	62.2	64.3
R3D-50 + STAM	65.3	62.9	63.1	63.8

Method	Top-1		
	Verb	Noun	Action
TSM [28]	63.2	48.0	36.2
I3D [9]	55.5	43.3	29.3
R3D-50 [10]	56.7	45.3	31.5

TSM + STAM	64.3	50.3	38.0
I3D + STAM	57.4	45.2	31.7
R3D-50 + STAM	58.0	47.4	33.7

Table 3: Results on the EGTEA dataset (left) and EPIC-100 dataset (right). * indicates optical flow is used, dark rows indicate gaze information is used. All experiments of our STAM only take RGB frames as inputs.

Quantitative result comparison is shown in Table 2. Using the same backbone and the same number of input clips, our STAM significantly outperforms other feature aggregation methods. This strongly validates that our STAM can effectively leverage the local and global information in the clip features to predict better temporal attention weights.

4.3 Comparison with SOTA

We conduct experiments to validate the improvement in performance that STAM brought to the backbones and compare the performance with other state-of-the-art methods. We compare the following methods for first-person action recognition: Ego-RNN [19], LSTA [10] and SAP [15]. We also add our proposed STAM on 3 commonly used backbones: TSM [28], I3D [9], and 3D-ResNet-50 (R3D-50) [10]. For the 2D backbone TSM we input multiple frames, and for other backbones the inputs are multiple 16-frame video clips.

Table 3-left and Table 3-right show the result comparison on the EGTEA dataset and the EPIC-100 dataset, respectively. From both tables, we can see that the performance of all backbones can be improved when adding our proposed STAM. On the EGTEA dataset, our method achieves the state-of-the-art performance among all the methods with RGB frames as inputs, and is even better than LSTA [10] which also uses optical flow. Only Min *et al.* [5] outperforms our method, but they use optical flow and human gaze as additional information. Similarly, our method can boost the performance of all backbones on the EPIC-100 dataset. This performance increase is significant given the intrinsic difficulty of the EPIC-100 dataset (e.g. non-scripted, unbalanced label, large variety of action).

4.4 Exploration of STAM

For all temporal aggregation methods, the number of input clips (temporal receptive field) can have a huge influence on the final performance. We explore the performance of our STAM with different numbers of clips as inputs.

Table 4 shows the influence of different numbers of attention layers used in our STAM. We experiment with 3, 6 and 9 input clips. In both tables, Avg indicates direct averaging of all clip features, which serves as a baseline for understanding the usefulness of temporal attention. Gain emphasize the maximum performance gain with respect to the Avg method.

Clip Num	Avg	Num of global att layers					Gain
		0	1	2	3	4	
3	59.1	61.1	61.7	62.2	63.5	62.5	+4.4
6	62.3	64.3	65.1	65.6	65.1	64.9	+3.3
9	63.0	65.5	66.2	66.5	66.9	66.7	+3.9

Table 4: Analysis of the influence of input clip (temporal receptive field) and the number of stacked global attention layers. Experiments are done using I3D backbone on the EGTEA dataset split1.



Figure 3: Visualization of temporal attention score of a few samples of the EGTEA dataset when stacking different numbers of global attention layers. The highest temporal attention score in each layer is underlined. In the case of global attention layer 0, initialization using the self-attention method is used. The frame with green background indicates that the backbone model can predict the correct action class with this clip alone as input.

Usually, since using more clips brings richer information, the recognition accuracy tends to become higher.

From the table, we can see that with the help of STAM, the performance gain of a small temporal receptive field (3 clips) is more obvious. This from one side proves that our STAM can learn to highlight the most important clip feature. The table also suggests that stacking two layers (with 6 clips as inputs) or three layers (with 3 or 9 clips as inputs) of global attention works the best, and stacking four layers saturates the performance.

Figure 3 shows visualizations of several sample actions in the EGTEA dataset together with the temporal attention for each clip. We use one frame to represent each 16-frame clip in this Figure, showing the model performance when using 6 clips as inputs. In each global attention layer, the highest temporal attention score is marked with an underline. Frames with green boundaries indicate that only with the corresponding clip as input, the network can predict the correct action class. For all the samples, with the stack of global attention layer, we can see the temporal attention gradually shifts, giving the correct clip higher weights. For

example, in the action “take paper towel”, the initialization of global layer using self-attention (global layer 0) gives more attention to a background clip. This is reasonable since with self-attention on each clip’s feature, the network tends to assign high values to clips that have similar features. With the help of the global attention layer, the highest temporal attention goes to the 4th and 5th clip, which contains sufficient information for correct action recognition and therefore should be emphasized. This strongly supports that the use of global feature is important. After stacking more global attention layers, the temporal attention is refined to give more focus on the correct clips. Other samples of “turn off faucet” and “open fridge” show similar performance. This proves that the proposed stacking of the global attention layers is effective in refining the temporal attention. In this Figure, the temporal attention score becomes optimal when stacking 2 global attention layers, and saturates if we stack more than three layers, showing a similar trend with Table 4.

4.5 Experiments on third-person dataset

Furthermore, to see how the proposed STAM performs on traditional third-person datasets, we add an experiment using the HMDB51 dataset [21]. HMDB51 is a widely used human motion dataset collected from YouTube that contains 6849 clips divided into 51 action categories. The average duration of each action is about 3s. Following the convention, we report the average result of three train/test splits.

With our proposed STAM, the performance of TSM (69.4 \rightarrow 70.3), I3D (73.1 \rightarrow 76.1) and 3D ResNet (67.8 \rightarrow 70.5) backbones are all improved. The improvement may not be as significant as in first-person videos, but it strongly proves the effectiveness of the proposed STAM, and demonstrates its generalization ability. Due to the page limit, the complete table of quantitative result comparison can be found in the supplementary material.

5 Conclusion

In this paper, we propose a Stacked Temporal Attention Module (STAM) that generates temporal attention to better aggregate clip features for first-person action recognition. Our STAM can be built on top of most existing models to improve their action recognition performance. Experiments on EPIC-100 and EGTEA datasets demonstrate that our STAM is capable of boosting the performance of multiple backbones. Experiments on the HMDB51 dataset suggests that our STAM also works on third-person videos. As for our future work, we plan to explore other design choices for constructing the global feature, and leveraging the stacked temporal attention on other video tasks.

Acknowledgement

This work was supported by JST AIP Acceleration Research Grant Number JPMJCR20U1 and JSPS KAKENHI Grant Number JP20H04205, Japan.

References

- [1] 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.
- [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] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.
- [4] Ali Diba, Vivek Sharma, and Luc Van Gool. Deep temporal linear encoding networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2329–2338, 2017.
- [5] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.
- [6] Alireza Fathi, Yin Li, and James M Rehg. Learning to recognize daily actions using gaze. In *European Conference on Computer Vision*, pages 314–327. Springer, 2012.
- [7] 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.
- [8] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *arXiv preprint arXiv:2106.02036*, 2021.
- [9] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.
- [10] 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.
- [11] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8401–8408, 2019.
- [12] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [13] Yifei Huang, Minjie Cai, Zhenqiang Li, and Yoichi Sato. Predicting gaze in egocentric video by learning task-dependent attention transition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 754–769, 2018.

- [14] Yifei Huang, Minjie Cai, Zhenqiang Li, Feng Lu, and Yoichi Sato. Mutual context network for jointly estimating egocentric gaze and action. *IEEE Transactions on Image Processing*, 29:7795–7806, 2020.
- [15] Yifei Huang, Minjie Cai, and Yoichi Sato. An ego-vision system for discovering human joint attention. *IEEE Transactions on Human-Machine Systems*, 50(4):306–316, 2020.
- [16] Yifei Huang, Yusuke Sugano, and Yoichi Sato. Improving action segmentation via graph-based temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14024–14034, 2020.
- [17] Georgios Kapidis, Ronald Poppe, Elsbeth van Dam, Lucas Noldus, and Remco Veltkamp. Multitask learning to improve egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.
- [18] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501, 2019.
- [19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [20] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6232–6242, 2019.
- [21] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011.
- [22] Jun Li, Xianglong Liu, Wenxuan Zhang, Mingyuan Zhang, Jingkuan Song, and Nicu Sebe. Spatio-temporal attention networks for action recognition and detection. *IEEE Transactions on Multimedia*, 22(11):2990–3001, 2020.
- [23] Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6943–6953, 2021.
- [24] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 287–295, 2015.
- [25] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 619–635, 2018.
- [26] Yin Li, Miao Liu, and Jame Rehg. In the eye of the beholder: Gaze and actions in first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [27] Zhenyang Li, Kirill Gavriluk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018.

- [28] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.
- [29] Minlong Lu, Danping Liao, and Ze-Nian Li. Learning spatiotemporal attention for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.
- [30] Minghuang Ma, Haoqi Fan, and Kris M Kitani. Going deeper into first-person activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1894–1903, 2016.
- [31] Kyle Min and Jason J Corso. Integrating human gaze into attention for egocentric activity recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1069–1078, 2021.
- [32] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.
- [33] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.
- [34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alche-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.
- [35] Wenjie Pei, Tadas Baltrusaitis, David MJ Tax, and Louis-Philippe Morency. Temporal attention-gated model for robust sequence classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6730–6739, 2017.
- [36] 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.
- [37] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1961–1970, 2016.
- [38] Suriya Singh, Chetan Arora, and CV Jawahar. First person action recognition using deep learned descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2620–2628, 2016.
- [39] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. *arXiv preprint arXiv:1807.11794*, 2018.

- [40] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9954–9963, 2019.
- [41] Maitreya Suin and AN Rajagopalan. Gated spatio-temporal attention-guided video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7802–7811, 2021.
- [42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [43] Jiaze Wang, Xiaojiang Peng, and Yu Qiao. Cascade multi-head attention networks for action recognition. *Computer Vision and Image Understanding*, 192:102898, 2020.
- [44] 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.
- [45] Xiaohan Wang, Linchao Zhu, Yu Wu, and Yi Yang. Symbiotic attention for egocentric action recognition with object-centric alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [46] Zachary Wharton, Ardhendu Behera, Yonghuai Liu, and Nik Bessis. Coarse temporal attention network (cta-net) for driver’s activity recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1279–1289, 2021.
- [47] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.
- [48] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2019.
- [49] 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.
- [50] Zheming Zuo, Longzhi Yang, Yonghong Peng, Fei Chao, and Yanpeng Qu. Gaze-informed egocentric action recognition for memory aid systems. *IEEE Access*, 6: 12894–12904, 2018.