Supplementary Materials for "Attention Distillation for Learning Video Representations"

Miao Liu[1]
mliu328@gatech.edu
Xin Chen[3]
xinchen@kwai.com
Yun Zhang[1]
yzhang467@gatech.edu
Yin Li[2]
yin.li@wisc.edu
James M. Rehg[1]
rehg@gatech.edu

[1] College of Computing
Georgia Institute of Technology
Atlanta, United States

[2] Departments of Biostatistics
and of Computer Sciences
University of Wisconsin-Madison
Madison, United States

[3] Y-tech
Kuaishou Technology
Palo Alto, United States

This is the supplementary material for our paper in BMVC 2020, titled 'Attention Distillation for Learning Video Representations". In this document, we introduce the implementation details. Moreover, we investigate the predicted attention maps and the learned features of our model and provide further analysis of our approach.

Network Architecture

We detail the network architecture of our full model (Prob-Atten) in Table 3. Specifically, our model adopts I3D network [1] as the backbone. We attached the attention modules to the last Inception Module of the fourth convolution block. The appearance and motion attention maps, predicted by attention modules, are further used to pool features from the last Inception Module of the fifth convolution block for classification. And their results are fused at the end for final recognition. The network takes 24 frames as inputs with dimension $24 \times 224 \times 224 \times 3$ (RGB). And the network outputs (a) a downsampled attention map with size $3 \times 7 \times 7$ (three temporal slices with spatial resolution of 7×7); and (b) the action scores for each category.

Analysis of Attention Distillation

We provide extensive analysis to understand what has been learned by our model. We show that these attention maps help to locate the spatial extents of actions. And, we study different approaches to evaluate whether the learned representation is sensitive to motion. Finally, we provide more visualization of the attention maps of our model.

Method	Prec	Recall	F1
Gaussian (center prior)	52.6	20.6	29.6
Saliency Map (DSS [2])	**51.2**	47.7	**49.4**
Soft-Atten (RGB)	33.8	40.5	36.9
Soft-Atten (Flow)	39.2	50.0	44.0
Our Appearance	31.5	52.1	39.2
Our Motion	36.3	**62.6**	46.0

Table 1: Results of action localization using attention maps on THUMOS'13 localization test set [3]. We report the best F1 score and its precision and recall. Our motion attention outperforms all baselines that are trained with only action labels.

Does the attention help to localize actions? We evaluate our output attention for action localization using THUMOS'13 localization dataset [3]–a subset of UCF101 with bounding box annotations for actions. We present our evaluation metric and discuss our results.

- **Evaluation Metric**. We consider action localization as binary labeling of pixels and report the F1 score from Precision-Recall (PR) curve. Specifically, we first rescale both attention maps and video frames into a fixed resolution (56×56). We then enumerate all thresholds and binarize the attention map. Each threshold defines a point on the PR curve. Given a binary attention map, a positive pixel is considered as a true positive if it is inside the bounding box, or it is within 10-pixel "tolerance zone" of the box. This tolerance is added to compensate for the reduced resolution of the attention map, as in [6]. We report the best F1 score on the curve and its corresponding precision and recall.

- **Results**. We compare attention maps from our model to a set of baseline methods, including a fixed Gaussian distribution (center prior), a latest deep saliency model (DSS [2]), and our Soft-Atten (RGB/Flow). The results are shown in Table 1. Our appearance attention beats the baselines of center prior and Soft-Atten (RGB), but is worse than Soft-Atten (flow). Our motion attention achieves the highest score among all methods that only receive action labels as supervision, and only under-performs DSS. We have to emphasis that directly comparing our results to DSS is unfair. DSS is trained with pixel-level annotations using external data and runs at the original video resolution, while our attention maps are trained using clip-level action labels and down-sampled both spatially (32x) and temporally (8x). These results suggest that our attention maps help to locate the spatial extent of actions.

Does our method learn better motion representation? We further study how the temporal order of the input video frames will affect the recognition performance. We conduct an experiment of classifying reverted videos as in [7, 8]. Specifically, we invert the frame order for all testing videos of UCF101 and HMDB51. We compare their recognition results with those from normal temporal order. If a model truly rely on motion representation for the recognition, this inversion will significantly decrease the recognition performance. We test the vanilla I3D RGB and flow models, as well as our model. And the results are presented in Table 2. Not surprisingly, I3D flow model has the largest performance drop. In contrast, I3D RGB is barely affected by the reverted arrow of time. Our model has a performance drop that is larger than I3D RGB yet much smaller than I3D flow. This is consistent with our results on action recognition. Our model does not capture the same level of motion information as the flow network.

How is the motion encoded? It is also possible that our model simply copies the motion attention map without encoding motion in the network. To eliminate this hypothesis, we

Dataset	Method	Mean Class Accuracy		
		Original	Reverted	DeltaΔ
UCF101	I3D RGB	94.8	94.7	0.1
	I3D flow	94.0	89.9	**4.1**
	Ours	95.7	95.1	0.6
HMDB51	I3D RGB	70.9	70.2	0.7
	I3D flow	73.9	66.0	**7.9**
	Ours	72.0	70.6	1.4

Table 2: Inverting the arrow of time for action recognition. We train the models on normal samples, yet test them on videos with reversed temporal order. A large performance drop indicates that the model has to rely on motion information for the recognition.

experimented with training an RGB network that directly combines a reference motion attention map and its own appearance attention map for action recognition. The reference motion attention is produced by a flow network during both training and testing. And the rest of this network follows exactly the same architecture as our model. This model has an accuracy of 95.1%/71.6% on UCF101/HMDB51, under-performing our model by -0.6%/-0.4% on UCF101/HMDB51. These results indicate that the distillation process not only generates motion attention maps, but also learns motion-aware representation.

Additional Visualizations. We provide additional visualization of attention maps in Fig 1. The figure follows the same format as Fig. 2 in our paper. These results further verify that (1) the appearance and motion attention maps are qualitatively different and (2) these attention map at good at localizing the actions, e.g., the actors or the moving regions.

What has been learned? Our visualization and action localization experiment suggest that our model learns to locate moving regions from video frames. However, when we invert the temporal order of frames, our learned features are not as sensitive as those from flow network. These results illustrate a key challenge for learning motion-aware representations. How the model learns to *identify* moving regions is not necessarily the right representation to *encode* motion. This is the same pitfall faced by our work and many previous work [4, 5]. And this challenge remains open.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[2] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. In *CVPR*, 2017.

[3] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *CVIU*, 2017.

[4] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, 2019.

[5] Joe Yue-Hei Ng, Jonghyun Choi, Jan Neumann, and Larry S Davis. Actionflownet: Learning motion representation for action recognition. In *WACV*, 2018.

[6] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015.

[7] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.

[8] Bolei Zhou, Alex Andonian, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.

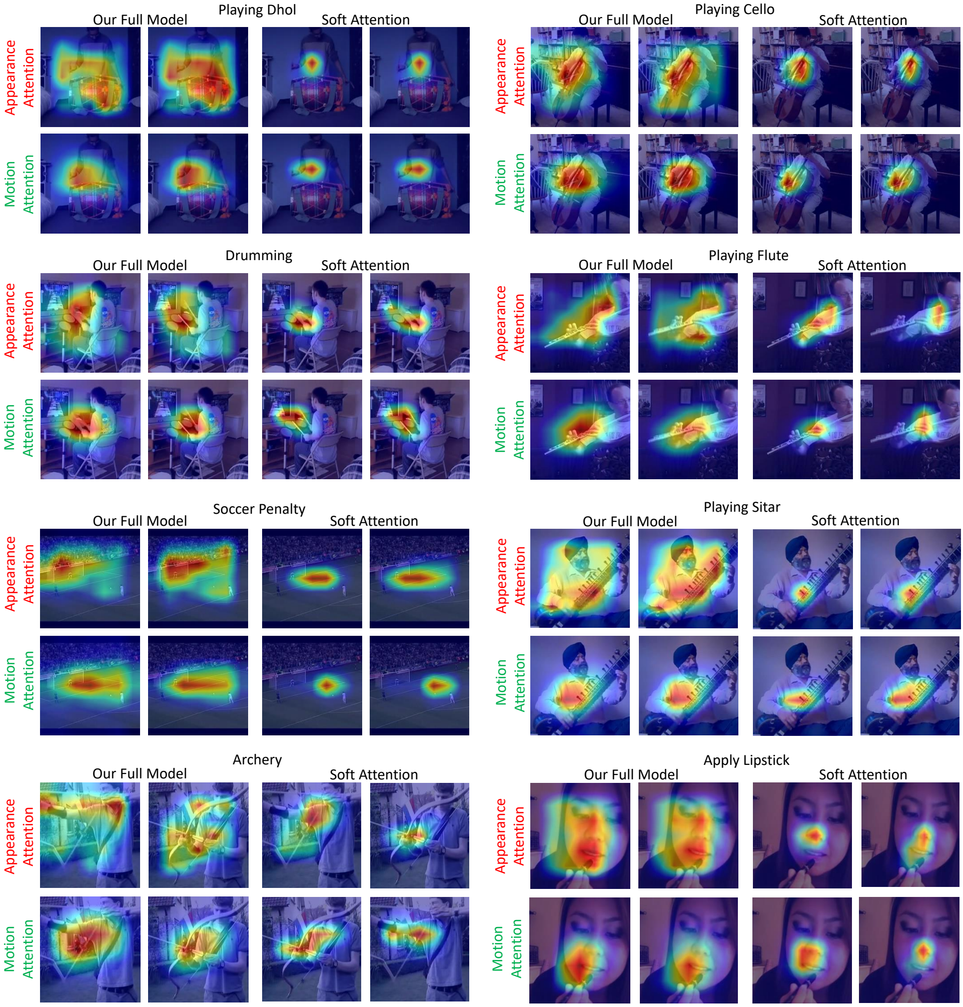

Figure 1: Visualization of attention maps from our full model and Soft-Atten. For each 24 frames video clip, we plot the attention heatmap over the first frame and last frame. Our model produces qualitatively different appearance and motion attention maps. And these attention maps are better at localizing the actions when compared to vanilla soft attention.

ID	Branch	Type	Kernel Size (THW)	Stride (THW)	Output Size (THWC)	Depth	Comments (Loss)
1	Backbone (shared)	Convolution	7x7x7	2x2x2	12x112x112x64	1	
2		Max Pool	1x3x3	1x2x2	12x56x56x64	0	
3		Convolution	1x1x1	1x1x1	12x56x56x64	1	
4		Convolution	3x3x3	1x1x1	12x56x56x192	1	
5		Max Pool	1x3x3	1x2x2	12x28x28x192	0	
6		Inception 3a			12x28x28x256	2	
7		Inception 3b			12x28x28x480	2	
8		Max Pool	3x3x3	2x2x2	6x14x14x480	0	
9		Inception 4a			6x14x14x512	2	
10		Inception 4b			6x14x14x512	2	
11		Inception 4c			6x14x14x512	2	
12		Inception 4d			6x14x14x528	2	
13		Inception 4e			6x14x14x832	2	Branching
14	Motion Attention Branch	Max Pool (on Inception 4e)	2x3x3	2x2x2	3x7x7x832	0	
15		Convolution	1x3x3	1x1x1	3x7x7x128	1	
16		Convolution	1x1x1	1x1x1	3x7x7x1	1	KL Loss (Attention Distillation)
17		Gumbel Softmax (Sampling)			3x7x7x1	0	Sampling Attention Map
18	Appearance Attention Branch	Max Pool (on Inception 4e)	2x3x3	2x2x2	3x7x7x832	0	
19		Convolution	1x3x3	1x1x1	3x7x7x128	1	
20		Convolution	1x1x1	1x1x1	3x7x7x1	1	KL Loss (Regularization)
21		Gumbel Softmax (Sampling)			3x7x7x1	0	Sampling Attention Map
22	Motion Action Branch	Max Pool (on Inception 4e)	2x2x2	2x2x2	3x7x7x832	0	
23		Inception 5a			3x7x7x832	2	
24		Inception 5b			3x7x7x1024	2	
25		Weighted Avg Pool	2x7x7	1x1x1	2x1x1x1024	0	Weights from Gumbel Softmax (Motion Attention Map)
26		Fully Connected			2x1x1x101	1	
27		Avg Pool	2x1x1	1x1x1	1x1x1x101	0	
28		Softmax			1x1x1x101	0	Cross Entropy Loss (Action Recognition)
29	Appearance Action Branch	Max Pool (on Inception 4e)	2x2x2	2x2x2	3x7x7x832	0	
30		Inception 5a			3x7x7x832	2	
31		Inception 5b			3x7x7x1024	2	
32		Weighted Avg Pool	2x7x7	1x1x1	2x1x1x1024	0	Weights from Gumbel Softmax (Appearance Attention Map)
33		Fully Connected			2x1x1x101	1	
34		Avg Pool	2x1x1	1x1x1	1x1x1x101	0	
35		Softmax			1x1x1x101	0	Cross Entropy Loss (Action Recognition)

Table 3: Network Architecture of our full model (Prob-Atten). The network attaches appearance and motion attention moduels to a backbone I3D network. We list details of all operations in the network, as well as where the loss functions are attached. Note that the predicted scores from Motion Action Branch and Appearance Action Branch are fused at the end for final recognition.