Attentive Action and Context Factorization

Yang Wang[1]
wang33@cs.stonybrook.edu
Vinh Tran[14]
tquangvinh@cs.stonybrook.edu
Gedas Bertasius[2]
gberta@seas.upenn.edu
Lorenzo Torresani[3]
lt@dartmouth.edu
Minh Hoai[14]
minhhoai@cs.stonybrook.edu

[1] Stony Brook University
Stony Brook, NY 11794, USA
[2] University of Pennsylvania
Philadelphia, PA 19104, USA
[3] Dartmouth College
Hanover, NH 03755, USA
[4] VinAI Research
Hanoi, Vietnam

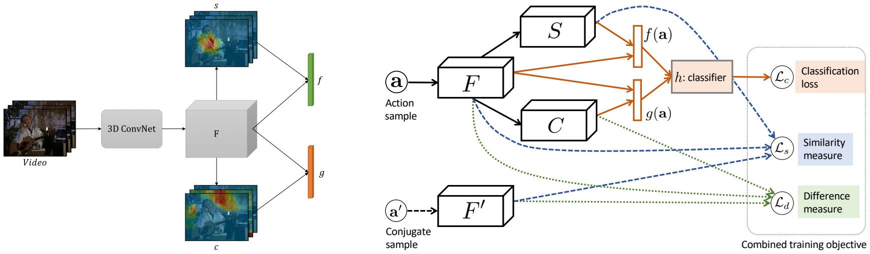
Abstract

We propose a method for human action recognition, one that can localize the spatiotemporal regions that 'define' the actions. This is a challenging task due to the subtlety of human actions in video and the co-occurrence of contextual elements. To address this challenge, we utilize conjugate samples of human actions, which are video clips that are contextually similar to human action samples but do not contain the action. We introduce a novel attentional mechanism that can spatially and temporally separate human actions from the co-occurring contextual factors. The separation of action and context factors is weakly supervised, eliminating the need for laboriously detailed annotation of these two factors in training samples. Our method can build human action classifiers with higher accuracy and better interpretability. Experiments on several human action recognition datasets demonstrate the quantitative and qualitative benefits of our approach.

1 Introduction

Our work is concerned with human actions [4, 10, 18, 20, 28, 35, 36, 38, 41, 46] in video, and we consider the task of localizing the spatiotemporal regions of a video that define a human action. This task is useful for understanding human actions and for developing computational models of human actions with better accuracy and higher interpretability. This task, however, is very challenging due to the subtlety of human actions and the co-occurrence of other contextual elements in the video. Finding the constituent elements of a human action requires more than grounding the decisions of a human action classifier [3, 30, 33, 47, 48, 50], because the decisions of a human action classifier might be partially based on some contextual cues [6, 37, 39, 40, 45] such as the background scene and the camera motion.

What we actually need is a detector rather than a classifier, and theoretically, this detector can be learned if there is training data in which the video voxels that define a human action are delineated. Unfortunately, collecting manual annotation at this level of details is notoriously difficult and even impossible. Scalability of manual annotation procedure is obviously



(a) test time behavior (b) learning framework

Figure 1: **Attentive Action and Context Factorization.** Attentive feature extraction and classifier learning are two stages of a joint learning framework. $S(\mathbf{a})$ and $C(\mathbf{a})$ are the probability maps for the action and context components. $f(\mathbf{a})$ and $g(\mathbf{a})$ are the action and context feature vectors, which will be subsequently fed into the classifier h. The objective is to minimize the classification loss, the similarity between action elements across F and F' and the difference between context elements across F and F'.

a major concern, but a bigger challenge comes from the inherent ambiguity of human action. In a video, which voxels define the action, which are part of the context, and which are noise? It is likely that we will get highly inconsistent annotations from different annotators.

In this paper, we propose a weakly-supervised method for localizing the spatiotemporal regions of a video that define a human action. We eliminate the need for laboriously and ambiguously detailed annotation by leveraging conjugate samples [42] of human actions, which are video clips that are contextually similar to human action samples, but do not contain the actions. We introduce a novel attentional mechanism that can spatially and temporally separate human actions from the co-occurring factors, and this attentional mechanism can be learned with the help of conjugate samples. Given an input video, this attentional mechanism will compute two spatiotemporal attention maps, one for the action components and one for the context components, as illustrated in Fig. 1(a). Using these attention maps, we can identify and separate action voxels from context voxels. This allows us to selectively pool information from relevant regions of a video to compute feature vectors for the action and context parts of the video. This leads to an improved classifier with higher accuracy and better interpretability.

2 Related Work

2.1 Visual grounding

Given a query phrase or a referring expression, visual grounding requires a model to specify a region within the image or the video that corresponds to the query input. It is inspired by the top-down influences on selective attention in the human visual system (see [1] for a review). Various methods [3, 33, 47, 48, 50] have been proposed for grounding a CNN's prediction for images. However, visual grounding for video data or temporal network architectures is much less explored. Karpathy et al. [21] visualized LSTM cells that keep track of long-range dependencies in a character-based model. Selvaraju et al. [30] presented qualitative results on grounding image captioning and visual question answering using an RNN. Bargal et al. [2] proposed a top-down attention mechanism for CNN-RNN models to produce spatiotemporal saliency maps that can be used for action/caption localization in videos.

2.2 Disentanglement in image/video synthesis

In recent years, much effort [7, 24, 26, 31, 32] has been spent on the task of learning explicitly disentangled representations and subsequently leveraging them to control image or video synthesis process. Early approaches used bilinear model [34] and encoder-decoder network [19] for separating content and style for images such as faces and text in various fonts. The disentanglement of content and style was further explored in [8, 23] for computer graphics applications. More recently, deep learning frameworks based on Variational Auto Encoders [22, 27] and Generative Adversarial Networks [5, 13, 29] have become more popular because they are more powerful than classical methods at encoding and synthesizing images and videos. Besides the disentanglement of content and style, some research [16, 32] has also been conducted on separating the feature representation into translation-related and translation-invariant factors.

Most related to our work is the Action-Context Factorization (ACF) framework [42] for training a human action classifier that can explicitly factorize human actions from the co-occurring context. However, the objective in ACF was to learn factorized feature extraction networks for better classification performance adaptability, while our goal here is to perform visual grounding of the two factors by automatically identifying the voxels associated to action and those corresponding to context. Incorporated in our method is a novel attentional mechanism to spatially and temporally separating human action from co-occurring context, a task that cannot be achieved by the ACF framework [42].

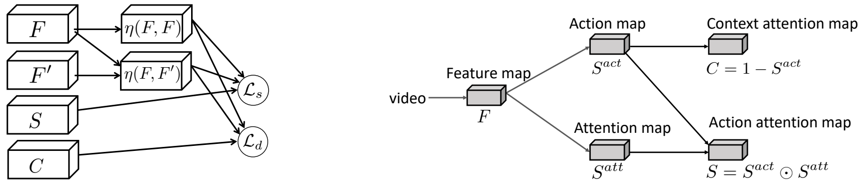
3 Attentive Factorization

3.1 Proposed Framework

We propose to train a human action classifier with both action and conjugate samples of human actions. Let the training data be $\{(l_i, \mathbf{a}_i, \mathbf{a}'_i)\}_{i=1}^n$, where l_i is an action label, $\mathbf{a}_i$ is the video sample for action l_i, and $\mathbf{a}'_i$ is a conjugate sample for $\mathbf{a}_i$. Video $\mathbf{a}'_i$ is contextually similar to $\mathbf{a}_i$ but it does not contain the action l_i.

Fig. 1(b) illustrates the proposed framework. Given the feature map $F = F(\mathbf{a}) \in \Re^{T\times H\times W\times D}$ for an action sample $\mathbf{a}$, we will compute two attention maps S and $C \in [0,1]^{T\times H\times W}$ for the spatiotemporal distribution of the action and the context elements of the video respectively. S_i is the probability that voxel i of the feature map F corresponds to an action component, while C_i is the probability that voxel i is a context element. Finally, the action extractor $f(\mathbf{a})$ can be defined as the function of the map F and the action map S, while the context extractor $g(\mathbf{a})$ is the function of F and the context map C. To aid the factorization process, our framework uses a conjugate sample $\mathbf{a}'$ and minimize two additional losses L_s and L_d. L_s measures the similarity between the action elements of $\mathbf{a}$ (defined based on S) and the action elements of $\mathbf{a}'$. Similarly, L_d measures the difference between the context elements of $\mathbf{a}$ and $\mathbf{a}'$, defined based on C. The combined training objective is to minimize:

$$\sum_{i=1}^{n} \mathcal{L}_c(h(f(\mathbf{a}_i), g(\mathbf{a}_i)), l_i) + \sum_{i=1}^{n} \left[\mathcal{L}_d(C(\mathbf{a}_i), F(\mathbf{a}_i), F(\mathbf{a}'_i)) + \mathcal{L}_s(S(\mathbf{a}_i), F(\mathbf{a}_i), F(\mathbf{a}'_i))\right]. \quad (1)$$



(a) Comparing weakly-aligned feature maps (b) Computing factorized attention maps

Figure 2: (a) To compare similarities and differences between weakly-aligned feature maps (F and F'), we first transform them, then compute the weighted average of the similarities and differences. The weights are based on the action map S and context map C. (b) illustrates the computation of action and context attention maps. F is a 4D tensor $F \in \Re^{T \times H \times W \times D}$, while S^{act}, S^{att}, S, C are 3D tensors, $\in [0,1]^{T \times H \times W}$.

This formulation uses conjugate samples in the learning process to aid the factorization between action and context components $f(\mathbf{a})$ and $g(\mathbf{a})$. This task is difficult or even impossible to achieve using the standard approach to train the classifier by minimizing the classification loss only. One can also train $f(\mathbf{a})$ and $g(\mathbf{a})$ with detailed annotations, but collecting detailed annotation is a laborious and ambiguous process. The above formulation eliminates the need for detailed annotations by utilizing conjugate samples during training. During testing, conjugate samples are not needed.

There are three loss terms in the above objective function: $\mathcal{L}_c, \mathcal{L}_s$, and $\mathcal{L}_d$. One technical challenge that we will address is the design of proper loss functions for $\mathcal{L}_s$ and $\mathcal{L}_d$. While $\mathcal{L}_c$ can be defined based on any standard classification loss such as cross entropy, the design of $\mathcal{L}_s$ and $\mathcal{L}_d$ requires special consideration because there is no direct voxel-to-voxel correspondence between an action sample and a conjugate sample. This problem will be addressed in the next subsection. After that, we will describe the parameterization of the action and context attention maps.

3.2 Comparing weakly-aligned feature maps

We now describe the loss functions for measuring the similarity and difference between weakly-aligned action and conjugate video samples.

In general, it is not a good idea to measure the element-wise similarities and differences between $F = F(\mathbf{a})$ and $F' = F(\mathbf{a}')$ due to the positional shift of action and context elements. Furthermore, an action sample and conjugate sample might have different lengths so the feature maps might even have different sizes. To address the problem of not having voxel-to-voxel correspondences, we propose the following mechanism. Consider a feature vector F_k at location k of the feature map F for the action sample. If F_k corresponds to an action element, it should be different from F'_j for all j's. On the other hand, if F_k corresponds to a context voxel, then there must be j such that F'_j is similar to F_k. We propose to define a quantity to measure the amount of feature vector F_k in a set of feature vectors F' as follows:

$$\eta(F_k, F') = \sum_j \frac{\exp(\gamma \hat{F}_k^T \hat{F}'_j)}{\sum_l \exp(\gamma \hat{F}_k^T \hat{F}'_l)} F'_j. \tag{2}$$

where $\hat{F}_k$ and $\hat{F}'_j$ are unit vectors. Similarly, we can compute a vector $\eta(F_k, F)$ to represent the amount F_k in F. $\eta(F_k, F)$ and $\eta(F_k, F')$ should be different or similar depending on whether F_k corresponds to an action or context voxel. We define the similarity and difference loss functions based on the transformed feature maps as follows:

$$\mathcal{L}_s = \frac{\sum_k S_k \times sim(\eta(F_k, F),\ \eta(F_k, F'))}{\sum_k S_k}, \tag{3}$$

$$\mathcal{L}_d = \frac{\sum_k C_k \times diff(\eta(F_k, F),\ \eta(F_k, F'))}{\sum_k C_k}. \tag{4}$$

Here, *sim* and *diff* are two functions that measure the similarity and difference between two vectors based on their cosine distance. Specifically in our experiments, $sim(\mathbf{x}, \mathbf{z}) = \max\left\{0,\ \cos\left\langle \mathbf{x}, \mathbf{z} \right\rangle - \xi\right\}$ and $diff(\mathbf{x}, \mathbf{z}) = 1 - cos\langle \mathbf{x}, \mathbf{z} \rangle$. Fig. 2(a) illustrates the procedure for feature map transformation and loss computation.

3.3 Attentive Action & Context Factorization

A video is represented by a 4D feature map $F \in \Re^{T \times H \times W \times D}$, and we will compute two attention maps S and $C \in [0,1]^{T \times H \times W}$ for the spatiotemporal distribution of the action and the context elements of the video. We parameterize S and C based on one action map $S^{act} \in [0,1]^{T \times H \times W}$ and one attention map $S^{att} \in [0,1]^{T \times H \times W}$ as illustrated in Fig. 2(b) and described below:

$$S = S^{act} \odot S^{att},\ C = 1 - S^{act}. \tag{5}$$

Operator $\odot$ denotes element-wise multiplication. The action map S^{act} is the output of a per-location sigmoid function. Its values represent the probability of each location being an action component. Conversely, $1 - S^{act}$ represents the probability of each location being a contextual element that is common in both action and conjugate samples. We use S^{att} to further refine the action attention map S, because not all action elements are equal for the action recognition task. S^{att} is a unit-sum tensor that allocates the percentage of attention for each location. It is produced by a softmax function instead of a sigmoid function, and should attend to more discriminative action components. More details on the parameterization of S^{act} and S^{att} are in the supplementary material.

With the convolutional feature map $F \in \Re^{T \times H \times W \times D}$ that represents the video, and two spatiotemporal attention maps S and $C \in [0,1]^{T \times H \times W}$ respectively for the action and the context elements, we propose to compute the action feature and the context feature as follows.

$$F_s = \frac{1}{\sum s_i} \sum_i s_i \cdot \theta(F)_i \text{ and } F_c = \frac{1}{\sum c_i} \sum_i c_i \cdot \theta(F)_i.$$

F_s and F_c are the action and the context feature representations for input video $\mathbf{v}$. θ is a learnable transformation that is applied to the original feature maps. In this work, we define $\theta(F) = relu(F + conv3d(F))$, which includes a residual connection to aid the optimization process [15]. Other transformation functions are also plausible. Finally, we apply the classifier h to obtain the class confidence scores. Specifically we concatenate F_s and F_c and apply a fully-connected layer.

3.4 Implementation Details

We now provide more details on the parameterization of S^{act} and S^{att} in Eq. (5). We first apply two separate 3D convolutional layers ϕ and ψ on the extracted feature map F to compute the action confidence map $\phi(F)$ and the attention confidence map $\psi(F)$. These are unnormalized confidence scores, and their range highly depends on the values of the convolution kernel weights of ϕ and ψ. To ensure consistent value ranges across different videos and time steps, we apply a normalization step, called $\operatorname*{AttNorm}_{dims}$. This function can be applied on a multidimensional input tensor; it will subtract the average value from the input tensor and then divide the result by the standard deviation. Both the average value and the standard deviation are computed along the specified dimensions $dims$. This $\operatorname*{AttNorm}_{dims}$ function is similar to a Batch-Norm layer with a key difference. The proposed normalization is performed among different spatial locations at different time steps, instead of different samples in a single batch. We also use a $\operatorname*{Percent}_{\rho}$ module to control the percentage of positive values to be $\rho \in [0, 100]$ in each confidence map. This can be done by subtracting the ρ-percentile value from every element of the confidence map. By explicitly controlling the percentage of positive values before the sigmoid function, as used in Eq. (6), we can regulate the percentage of action components in S^{act} within the entire video. Both $\operatorname*{AttNorm}_{dims}$ and $\operatorname*{Percent}_{\rho}$ are parameter-free. They enable efficient and stable network training in our experiments. After the normalization of the confidence scores, we apply the relaxed sigmoid or softmax function to obtain the intermediate attentive distribution maps:

$$S^{act} = \text{sigmoid}(\alpha \cdot \operatorname*{Percent}_{\rho} \operatorname*{AttNorm}_{t,h,w} \phi(F)), S^{att} = \operatorname*{softmax}_{t,h,w}(\beta \cdot \operatorname*{AttNorm}_{t,h,w} \psi(F)). \tag{6}$$

Here, ρ, α, and β are tunable hyper parameters. ρ controls the percentage of action component within each video. α controls the sharpness of the boundary between action and context. When α is 0, S^{act} is 0.5 everywhere, and every location contains equal amounts of action and context values. If $\alpha = \infty$, ρ percent of the values in S^{act} are 1's, and the rest are 0's. That entails a hard separation of action and context spatiotemporally. β can be used to control the balance between average pooling and max pooling for the attention map. When β is 0, S^{att} has uniform values, and this is equivalent to average pooling. If $\beta = \infty$, only one location in the video has the attention weight of 1, while the attention weights of other locations are 0; this is equivalent to max pooling.

The learnable parameters of the proposed action and context attentive mechanism are the kernel parameters of the functions ϕ, ψ and θ. The functions $\operatorname*{AttNorm}_{dims}$, $\operatorname*{Percent}_{\rho}$, sigmoid and softmax are all parameter-free. ρ, α and β are hyper-parameters which can be tuned using validation data; in our experiments, a good default starting point is $\rho = 30.0, \alpha = \beta = 1$.

4 Experiments

We perform experiments on four action recognition datasets: ActionThread [17], Hollywood2 [25], HACS [49], and Pascal VOC [9] (presented in supplementary material).

4.1 Separating action and context in video

Dataset. Our experiments in this section are performed on the ActionThread dataset [17]. It contains not only the video clips that include human actions but also the sequences right before and after the actions. These sequences are good candidates for conjugate samples of human actions. The human action samples in ActionThread were automatically located and extracted using script mining in 15 different TV series. ActionThread has 13 different actions and 3035 video clips, which are split into disjoint train and test subsets [17]. We consider the

Method	RGB Only		RGB + Flow	
	#param	mAP	#param	mAP
I3D [4]	13.4M	46.7	28.8M	55.9
PoseMask	n/a	46.0	n/a	53.2
10Region	n/a	50.1	n/a	57.6
50Region	n/a	49.6	n/a	56.3
Attentional [11]	13.4M	53.0	28.8M	61.8
ACF [42]	13.4M	52.9	28.8M	60.4
ActX [w/o S^{att}]	13.4M	51.2	28.8M	58.8
ActX [Proposed]	13.4M	**55.4**	28.8M	**63.1**

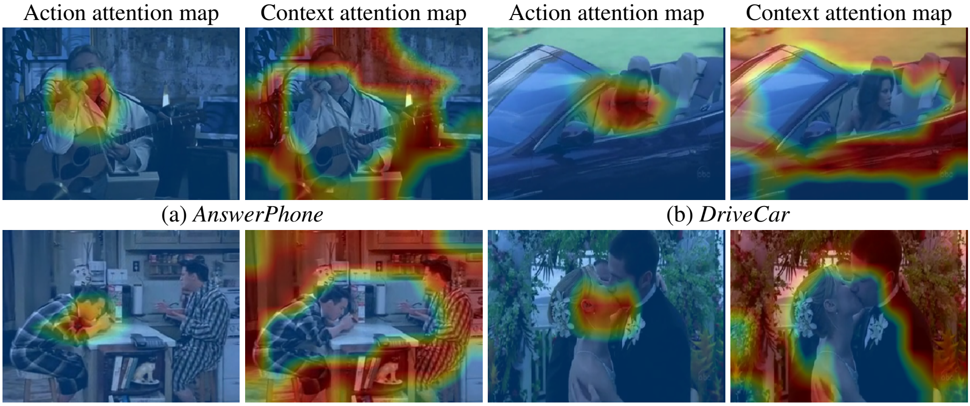
Table 1: **Action recognition results on the ActionThread dataset** (average precision values are shown; higher is better). All methods are based on the same feature maps produced by the I3D backbone. I3D [4] uses global average pooling. All other methods use attention for weighted pooling of features. The numbers of parameters added by the attentional mechanisms are negligible compared to that of the I3D backbone. ActX achieves the best performance.

pre- and post-action sequences as the source for mining conjugate samples, and make sure each pair of action and conjugate samples are extracted from the same video thread.

Feature Extraction. We use the pre-trained two-stream I3D ConvNet [4] for feature extraction. Each video is represented as two 4D feature maps $F_{rgb}, F_{flow} \in \Re^{T \times 8 \times 11 \times 1024}$. T is the number of video frames divided by 8. The proposed attentive factorization network can take input feature maps with different temporal lengths. However, temporal cropping with a fixed length is still necessary during training because we need to put feature maps into mini-batches. We use a batch size of 36 videos and the temporal cropping length is set to 10, equivalent to 80 frames or 3.2 seconds. At test time, temporal cropping is unnecessary. The entire feature maps are fed into the network to perform attentive action and context factorization as well as action recognition. More details are in the supplementary material.

Action Recognition Performance. Tab. 1 compares the action recognition performance of several methods on the ActionThread dataset. **I3D** is a baseline approach [4] where the I3D feature vectors are averaged over the entire video. **ActX** is the proposed approach where attentive action and context factorization and human action classification are jointly optimized. As can be seen from Tab. 1, the proposed ActX significantly outperforms I3D, a direct baseline method without an attentional mechanism. The improvement brought by the attentional mechanism is most evident on the spatial stream where RGB images are used for feature extraction.

We also compare our method with alternative approaches for weighting the I3D feature maps. **ACF** is the approach of [42] where $\mathcal{L}_{sim}^{action}$ and $\mathcal{L}_{diff}^{context}$ are defined based on cosine distance between globally pooled feature vectors. **Attentional** is a state-of-the-art attentional pooling method [11] where both the top-down (category-specific) and the bottom-up (category-agnostic) attention maps are learned for weighting the predicted action scores. **PoseMask** is a method where human poses are used to obtain an attentive action map that focuses on regions of the human body joints. We use the Convolutional Pose Machine algorithm [44] to detect human pose keypoints within each video frame. The detected body keypoints include ears, eyes, and mouth regions, as well as shoulders, hips, limbs, wrists, and



(a) *AnswerPhone* (b) *DriveCar*

(c) *Eat* (d) *Kiss*

Figure 3: **Examples of attention maps for action and context.** The action attention maps have higher weights on humans, but not all humans receive the same attention, and not all parts of a human subject receive attention. The weights of the context maps are lower on the human subjects. The context maps have nonuniform distribution over background pixels.

ankles. We convert these keypoint heat maps into binary masks and use them for weighted pooling of the I3D features. The pose mask is expected to guide the classifier to attend to the human regions and achieve better action recognition results. However, the performance of PoseMask is worse than the average-pooled I3D features, indicating the importance of non-human regions. Because of this, we also consider **k-Region** where object region proposals are used for attentive attribution. We use the Detectron software library [12] to obtain the top k object regions predicted by the Region Proposal Network of a Mask RCNN. We use the ROI-align module to obtain the feature representation for each region and average them into a single feature vector. We experiment with $k = 10$ and $k = 50$, both yielding an action recognition performance that is slightly better than the average pooling method (I3D). However, these methods are still outperformed by the proposed method ActX. We also perform an ablation study where S^{att} is not used (i.e., $S = S^{act}$). The performance is not as good as when S^{att} is used to refine the action attention map. The numbers of parameters added by these attentional mechanisms are negligible compared to that of the I3D backbone network.

Visualizing Action and Context Components. Fig. 3 shows some examples of action and context attention maps. To recognize an action, it is important to attend to the human subjects. This explains why action maps put higher weights on human subjects. However, not all humans in a video frame receive the same attention, and not all body parts of a human receive attention. On the contrary, the weights of the context maps are lower on the human subjects. The context maps emphasize the background regions with a nonuniform distribution.

4.2 Imperfect conjugate samples

Until now, we have assumed the availability of perfect conjugate samples that do not contain the actions in consideration. This assumption holds in general. For example, if a human action dataset has temporal annotations for the start and end times of the actions, we can consider the sequences right before and after the action boundaries as conjugate samples. When a new action dataset is being collected, we can save the pre- and post-action sequences

Method	RGB Only	RGB + Flow
I3D [4]	55.3	62.8
PoseMask	49.4	60.0
10Region	55.6	64.6
50Region	55.9	64.8
Attentional [11]	61.3	69.8
ACF [42]	61.3	71.2
ActX [Proposed]	**62.5**	**73.2**
EEP + I3D [43]	73.5	81.0
EEP + ActX [Proposed]	**76.0**	**82.0**

Table 2: **Action recognition results on the Hollywood2 dataset** (average precision values are shown; higher is better). All methods are based on the same feature maps produced by the I3D ConvNet. The proposed ActX achieves better performance than other attention mechanisms, when used with or without Eigen Evolution Pooling (EEP [43]). We consider EEP because it has the current state-of-the-art performance on Hollywood2.

for future mining of conjugate samples. However, if we are forced to work with a specific dataset such as Hollywood2, where the pre- and post-action sequences are not available, we can use "imperfect" conjugate samples instead. In this scenario, the conjugate samples are not guaranteed to exclude all action elements that are shared with the action samples. With this in mind, we investigate whether or not a black-and-white separation between conjugate and action samples with respect to the action elements are necessary.

Experiments on Hollywood2. We first perform experiments on the Hollywood2 dataset [25], which contains 12 action classes and 1,707 videos collected from 69 Hollywood movies. To train the proposed method on Hollywood2, we use a batch size of 36 videos, and the temporal cropping length for I3D feature maps is set to 6, equivalent to 48 frames or around 2 seconds. The source for mining conjugate samples is the sequences right before and after the temporally cropped action samples. The extracted action and conjugate samples share the same context, and the key difference between them is the dynamic content of the human action at different action stages.

Tab. 2 compares the action recognition performance of several methods on the Hollywood2 dataset. ActX is the proposed approach where attentive action and context factorization and human action classification are jointly optimized. It significantly outperforms the baseline I3D [4] approach where the I3D feature vectors are averaged. The performance gaps between the proposed ActX method and other attention mechanism on the Hollywood2 dataset are similar to those on the ActionThread dataset, presented in Tab. 1. More specifically, the proposed ActX significantly outperforms Attentional, where both the category-specific and agnostic attention maps are used, while PoseMask, *k*-Region achieve worse or similar action recognition performance than the average-pooled I3D features. The last two rows of Tab. 2 also compares the action recognition performance of the baseline I3D features and the proposed ActX features, when they are temporally compressed using Eigen Evolution Pooling (EEP) [43]. EEP uses a set of basis functions learned from data using PCA to encode the evolution of features over time. It provides an effective way to capture the long-term and complex dynamics of human actions in video. ActX still outperforms I3D when combined with EEP.

Experiments on HACS-30. We also perform an experiment on the HACS dataset [49]. This

Method	mAP	mAcc
I3D [14]	57.8	83.1
Attention [11]	58.6	84.5
ACF [42]	58.3	84.3
ActX [Proposed]	**59.1**	**85.3**

Table 3: **Average Precision results on the HACS-30 validation set.** All methods are based on the same feature maps produced by the backbone network (I3D-Res34-RGB [14]). The proposed ActX achieves the best performance.

dataset provides both positive and negative samples for 200 action categories. Some positive and negative samples are extracted from the same video, so they have similar context with the key difference being whether or not an action of interest is observed. Thus the negative samples are a great source for mining conjugate samples. More precisely, given a positive action sample, we consider its top three nearest neighbors in the negative data pool as the conjugate samples. To compute the distance between videos, we use feature embeddings extracted by the I3D [14] network (I3D-Res34-RGB).

Limited by computation resource, we use a subset of the HACS dataset, which we will refer to as HACS-30. This subset contains 30 positive action classes and one negative background class. The 30 actions are those with the least positive samples, which makes the recognition task more challenging. After conjugate sample mining, the training set of HACS-30 contains 11770 positive samples, and each positive sample has three corresponding conjugate samples. We evaluate the action recognition performance on the HACS-30 validation set, which contains 1186 samples for the 30 actions and 1712 samples for the negative background class. Given the imbalance between positive and negative data, we report the mean average precision (mAP) and the mean class accuracy (mAcc; obtained by averaging the per-class accuracies over 31 classes). The experiment results are reported in Tab. 3. As can be observed, the proposed ActX outperforms other attention mechanisms. However, the performance gain is smaller than those achieved on the ActionThread and Hollywood2 dataset. This might be due to the short duration of HACS video clips (all clips are only 2 seconds long), and the added benefit of an attention mechanism is less for short videos.

5 Conclusions

We have presented a novel method for identifying action and context voxels of a video. Our method disentangles the action and the context components of a video with a novel attentional mechanism that can compute two spatiotemporal attention maps for action and context separately. Our method requires paired training data of action and conjugate samples of human actions, but this type of data can be collected easily. We have demonstrated the quantitative and qualitative benefits of using our method on the ActionThread, Hollywood2, HACS, and VOC Action datasets.

Acknowledgement. This project is supported by the National Science Foundation Award IIS-1566248, a Google Research Award, the SUNY 2020 Infrastructure Transportation Security Center, and VinAI Research.

References

[1] Farhan Baluch and Laurent Itti. Mechanisms of top-down attention. *Trends in neurosciences*, 34 (4):210–224, 2011.

[2] Sarah Adel Bargal, Andrea Zunino, Donghyun Kim, Jianming Zhang, Vittorio Murino, and Stan Sclaroff. Excitation backprop for rnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[3] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the International Conference on Computer Vision*, 2015.

[4] 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*, 2017.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016.

[6] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ2-nets: Double attention networks. In *Advances in Neural Information Processing Systems*, 2018.

[7] Emily L Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, 2017.

[8] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[9] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. www.pascal-network.org/challenges/VOC/voc2012/workshop/, 2012.

[10] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems*, 2016.

[11] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems*, 2017.

[12] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*. 2014.

[14] 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*, 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[16] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, 2011.

[17] Minh Hoai and Andrew Zisserman. Thread-safe: Towards recognizing human actions across shot boundaries. In *Proceedings of the Asian Conference on Computer Vision*, 2014.

[18] Minh Hoai and Andrew Zisserman. Improving human action recognition using score distribution and ranking. In *Proceedings of the Asian Conference on Computer Vision*, 2014.

[19] Fu Jie Huang, Y-Lan Boureau, Yann LeCun, et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

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

[21] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. In *ICLR Workshop*, 2016.

[22] Diederik Kingma and Max Welling. Auto-encoding variational bayes. 2014.

[23] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, 2015.

[24] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, et al. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, 2017.

[25] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[26] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, 2016.

[27] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*, 2017.

[28] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the International Conference on Computer Vision*, 2017.

[29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434, 2015.

[30] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the International Conference on Computer Vision*, 2017.

[31] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[32] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proceedings of the European Conference on Computer Vision*, 2018.

[33] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014.

[34] Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. volume 12, pages 1247–1283. MIT Press, 2000.

[35] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the International Conference on Computer Vision*, 2015.

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

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[38] 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 *Proceedings of the European Conference on Computer Vision*, 2016.

[39] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision*, 2018.

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[41] Yang Wang and Minh Hoai. Improving human action recognition by non-action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[42] Yang Wang and Minh Hoai. Pulling actions out of context: Explicit separation for effective combination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[43] Yang Wang, Vinh Tran, and Minh Hoai. Eigen-evolution dense trajectory descriptors. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2018.

[44] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[45] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[46] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[47] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*, 2014.

[48] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. volume 126, pages 1084–1102. Springer, 2018.

[49] Hang Zhao, Zhicheng Yan, Lorenzo Torresani, and Antonio Torralba. Hacs: Human action clips and segments dataset for recognition and temporal localization. *arXiv preprint arXiv:1712.09374*, 2019.

[50] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.