

Excitation Backprop for RNNs

Sarah Adel Bargal^{*1}, Andrea Zunino^{*2}, Donghyun Kim¹, Jianming Zhang³,
Vittorio Murino^{2,4}, Stan Sclaroff¹

¹Department of Computer Science, Boston University ²Pattern Analysis & Computer Vision (PAVIS), Istituto Italiano di Tecnologia ³Adobe Research ⁴Computer Science Department, Università di Verona
{sbargal,donhk,sclaroff}@bu.edu, {andrea.zunino,vittorio.murino}@iit.it, jianmzha@adobe.com

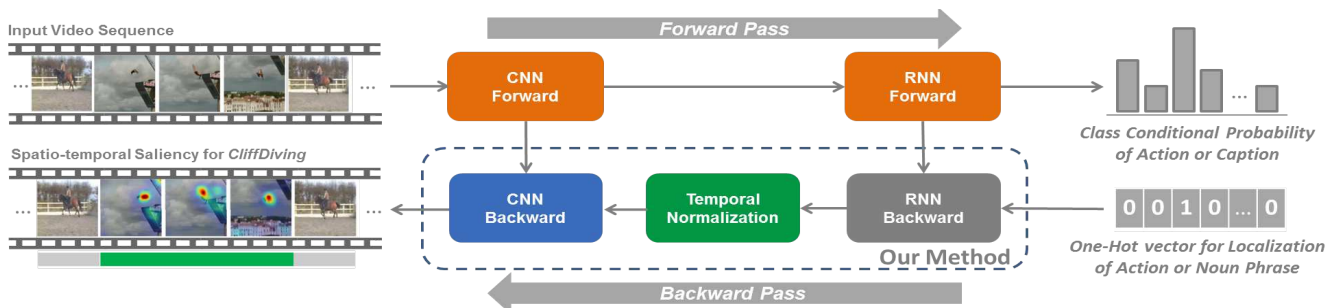


Figure 1: Our proposed framework spatiotemporally highlights/grounds the evidence that an RNN model used in producing a class label or caption for a given input video. In this example, by using our proposed back-propagation method, the evidence for the activity class *CliffDiving* is highlighted in a video that contains *CliffDiving* and *HorseRiding*. Our model employs a single backward pass to produce saliency maps that highlight the evidence that a given RNN used in generating its outputs.

Abstract

Deep models are state-of-the-art for many vision tasks including video action recognition and video captioning. Models are trained to caption or classify activity in videos, but little is known about the evidence used to make such decisions. Grounding decisions made by deep networks has been studied in spatial visual content, giving more insight into model predictions for images. However, such studies are relatively lacking for models of spatiotemporal visual content – videos. In this work, we devise a formulation that simultaneously grounds evidence in space and time, in a single pass, using top-down saliency. We visualize the spatiotemporal cues that contribute to a deep model’s classification/captioning output using the model’s internal representation. Based on these spatiotemporal cues, we are able to localize segments within a video that correspond with a specific action, or phrase from a caption, without explicitly optimizing/training for these tasks.

1. Introduction

To visualize what in a video gives rise to an output of a deep recurrent network, it is important to consider space and

time saliency, *i.e.*, where and when. The visualization of what a deep recurrent network finds salient in an input video can enable interpretation of the model’s behavior in action classification, video captioning, and other tasks. Moreover, estimates of the model’s attention (*e.g.*, saliency maps) can be used directly in localizing a given action within a video or in localizing the portions of a video that correspond to a particular concept within a caption.

Several works address visualization of model attention in Convolutional Neural Networks (CNNs) for image classification [1, 34, 24, 33, 21, 36, 18]. These methods produce saliency maps that visualize the importance of class-specific image regions (spatial localization). Analogous methods for Recurrent Neural Network (RNN)-based models must handle more complex recurrent, non-linear, spatiotemporal dependencies; thus, progress on RNNs has been limited to [8, 15]. Karpathy *et al.* [8] visualize the role of Long Short Term Memory (LSTM) cells for text input, but not for visual data. Ramanishka *et al.* [15] map words to regions in the video captioning task by dropping out (exhaustively or by sampling) video frames and/or parts of video frames to obtain saliency maps. This can be computationally expensive, and does not consider temporal evolution but only frame-level saliency.

^{*}Equal contribution

In contrast, we propose the first one-pass formulation for visualizing spatiotemporal attention in RNNs, without selectively dropping or sampling frames or frame regions. In our proposed approach, *contrastive* Excitation Backprop for RNNs (cEB-R), we address how to ground¹ decisions of deep recurrent networks in space and time simultaneously, using top-down saliency. Our approach models the top-down attention mechanism of deep models to produce interpretable and useful task-relevant saliency maps. Our saliency maps are obtained implicitly without the need to re-train models, unlike models that include explicit attention layers [29, 30]. Our method does not require a model trained using explicit spatial (region/bounding box) or temporal (frame) supervision.

Fig. 1 gives an overview of our approach that produces saliency maps which enable us to visualize where and when an action/caption is occurring in a video. Given a trained model, we perform the standard forward pass. In the backward pass, we use cEB-R to compute and propagate winning neuron probabilities normalized over space and time. This process yields spatiotemporal attention maps. Our demo code is publicly available².

We evaluate our approach on two models from the literature: a CNN-LSTM trained for video action recognition, and a CNN-LSTM-LSTM (encoder-decoder) trained for video captioning. In addition, we show how the spatiotemporal saliency maps produced for these two models can be utilized for localization of segments within a video that correspond to specified activity classes or noun phrases.

In summary, our contributions are:

- We are the first to formulate top-down saliency in deep recurrent models for space-time grounding of videos.
- We do so using a *single contrastive* Excitation Backprop pass of an already trained model.
- Although we are not directly optimizing for localization (no training is performed on spatial or temporal annotations), we show that the internal representation of the model can be utilized to perform localization.

2. Related Work

Several works in the literature give more insight into CNN model predictions, *i.e.*, the *evidence* behind deep model predictions. Such approaches are mainly devised for image understanding and can identify the importance of class-specific image regions by means of saliency maps in a weakly-supervised way.

Spatial Grounding. Ribeiro *et al.* [16] explained classification predictions with applications on images. Fong *et al.* [3] addressed spatial grounding in images by exhaustively

perturbing image regions. Guided Backpropagation [24] and Deconvolution [33, 21] used different variants of the standard backpropagation error and visualized salient parts at the image pixel level. In particular, starting from a high-level feature map, [33] inverted the data flow inside a CNN, from neuron activations in higher layers down to the image level. Guided Backpropagation [24] introduced an additional guidance signal to standard backpropagation preventing backward flow of negative gradients. Simonyan *et al.* [21] directly computed the gradient of the class score with respect to the image pixel to find the spatial cues that help the class predictions in a CNN. CAM [36] removed the last fully connected layer of a CNN and exploited a weighted sum of the last convolutional feature maps to obtain the class activation maps. Zhang *et al.* [34] generated class activation maps from any CNN architecture that uses nonlinearities producing non-negative activations. Oquab *et al.* [13] used mid-level CNN outputs on overlapping patches, requiring multiple passes through the network.

Spatiotemporal Grounding. Weakly-supervised visual saliency is much less explored for temporal architectures. Karpathy *et al.* [8] visualized interpretable LSTM cells that keep track of long-range dependencies such as line lengths, quotes, and brackets in a character-based model. Li *et al.* [10] visualized a unit’s salience for NLP. Selvaraju *et al.* [18] qualitatively present grounding for captioning and visual question answering in images using an RNN. Ramanishka *et al.* [15] explored visual saliency guided by captions in an encoder-decoder model. In contrast, our approach models the top-down attention mechanism of CNN-RNN models to produce interpretable and useful task-relevant spatiotemporal saliency maps that can be used for action/caption localization in videos.

3. Background: Excitation Backprop

In this section, a brief background on Excitation Backprop (EB) [34] is given. EB was proposed for CNNs in that work. In general, the forward activation of neuron a_j in a CNN is computed by $\hat{a}_j = \phi(\sum_i w_{ij}\hat{a}_i + b_i)$, where \hat{a}_i is the activation coming from a lower layer, ϕ is a nonlinear activation function, w_{ij} is the weight from neuron i to neuron j , and b_i is the added bias at layer i . The EB framework makes two key assumptions about the activation \hat{a}_j which are satisfied in the majority of modern CNNs due to wide usage of the *ReLU* non-linearity: **A1.** \hat{a}_j is non-negative, and **A2.** \hat{a}_j is a response that is positively correlated with its confidence of the detection of specific visual features.

EB realized a probabilistic Winner-Take-All formulation to efficiently compute the probability of each neuron recursively using conditional winning probabilities $P(a_i|a_j)$, normalized $\hat{a}_i w_{ij}$ (Fig. 2). The top-down signal is a prior distribution over the output units. EB passes top-down signals through excitatory connections having non-negative

¹In this work we use the terms *ground* and *localize* interchangeably.

²<https://github.com/sbargal/Caffe-ExcitationBP-RNNs>

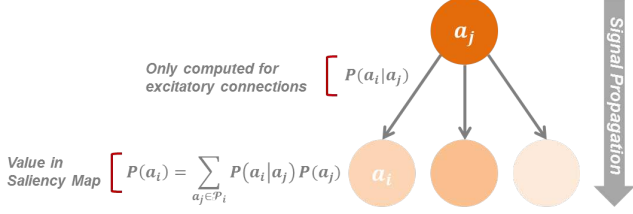


Figure 2: In Excitation Backprop, excitation probabilities are propagated in a single backward pass in the CNN. A top-down signal is a probability distribution over the output units. The probabilities are backpropagated from every parent node to its children through its excitatory connections. The figure illustrates the contributions of a single parent neuron to the excitation probabilities computed at the next layer. Each $P(a_i)$ in the saliency map is computed over the complete parent set \mathcal{P}_i . Shading of nodes in the figure conveys $P(a_i)$ (darker shade = greater $P(a_i)$).

weights, excluding from the competition inhibitory ones. Recursively propagating the top-down signal and preserving the sum of backpropagated probabilities layer by layer, it is possible to compute task-specific saliency maps from any intermediate layer in a single backward pass.

To improve the discriminativeness of the saliency maps, [34] introduced *contrastive* EB (cEB) which cancels out common winner neurons and amplifies the class discriminative neurons. To do this, given an output unit $o_i \in O$, a dual unit $\bar{o}_i \in \bar{O}$ is virtually generated, whose input weights are the negation of those of o_i . By subtracting the saliency map for \bar{o}_i from the one for o_i the result better highlights cues in the image that are unique to the desired class.

4. Our Framework

In this section we explain the details of our spatiotemporal grounding framework: cEB-R. As illustrated in Fig. 1, we have three main modules: RNN Backward, Temporal normalization, and CNN Backward.

RNN Backward. This module implements an excitation backprop formulation for RNNs. Recurrent models such as LSTMs are well-suited for top-down temporal saliency as they explicitly propagate information over time. The extension of EB for Recurrent Networks, EB-R, is not straightforward since EB must be implemented through the unrolled time steps of the RNN and since the original RNN formulation contains *tanh* non-linearities which do not satisfy the EB assumptions A1 and A2. [4, 6] have conducted an analysis over variations of the standard RNN formulation, and discovered that different non-linearities performed similarly for a variety of tasks. This is also reflected in our experiments. Based on this, we use *ReLU* nonlinearities and corresponding derivatives, instead of *tanh*. This satisfies A1 and A2, and gives similar performance on both tasks.

Working backwards from the RNN’s output layer, we compute the conditional winning probabilities from the set of output nodes O , and the set of dual output nodes \bar{O} :

$$P^t(a_i|a_j) = \begin{cases} Z_j \hat{a}_i^t w_{ij}, & \text{if } w_{ij} \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$\bar{P}^t(a_i|a_j) = \begin{cases} Z_j \hat{a}_i^t \bar{w}_{ij}, & \text{if } \bar{w}_{ij} \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

$Z_j = 1 / \sum_{i: w_{ij} \geq 0} \hat{a}_i^t w_{ij}$ is a normalization factor such that the sum of all conditional probabilities of the children of a_j (Eqn.s 1, 2) sum to 1; $w_{ij} \in W$ where W is the set of model weights and w_{ij} is the weight between child neuron a_i and parent neuron a_j ; $\bar{w}_{ij} \in \bar{W}$ where \bar{W} is obtained by negating the model weights at the classification layer only. $\bar{P}^t(a_i|a_j)$ is only needed for *contrastive* attention.

We compute the neuron winning probabilities starting from the prior distribution encoding a given action/caption as follows:

$$P^t(a_i) = \sum_{a_j \in \mathcal{P}_i} P^t(a_i|a_j) P^t(a_j) \quad (3)$$

$$\bar{P}^t(a_i) = \sum_{a_j \in \mathcal{P}_i} \bar{P}^t(a_i|a_j) \bar{P}^t(a_j) \quad (4)$$

where \mathcal{P}_i is the set of parent neurons of a_i .

Temporal Normalization. Replacing *tanh* non-linearities with *ReLU* non-linearities to extend EB in time does not suffice for temporal saliency. EB performs normalization at every layer to maintain a probability distribution. Hence, for spatiotemporal localization, signals from the desired n^{th} time-step of a T -frame clip should be normalized in both time and space (assuming S neurons in current layer) before being further backpropagated into the CNN:

$$P_N^t(a_i) = P^t(a_i) / \sum_{t=1}^T \sum_{i=1}^S P^t(a_i). \quad (5)$$

$$\bar{P}_N^t(a_i) = \bar{P}^t(a_i) / \sum_{t=1}^T \sum_{i=1}^S \bar{P}^t(a_i). \quad (6)$$

cEB-R computes the difference between the normalized saliency maps obtained by EB-R starting from O , and EB-R starting from \bar{O} using negated weights of the classification layer. cEB-R is more discriminative as it grounds the evidence that is unique to a selected class/word. For example, cEB-R of *Surfing* will give evidence that is unique to *Surfing* and not common to other classes used at training time (see Fig. 5 for an example). This is conducted as follows:

$$Map^t(a_i) = P_N^t(a_i) - \bar{P}_N^t(a_i). \quad (7)$$

CNN Backward. For every video frame f_t at time step t , we use the backprop of [34] for all CNN layers:

$$P^t(a_i|a_j) = \begin{cases} Z_j \hat{a}_i^t w_{ij}, & \text{if } w_{ij} \geq 0, \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$Map^t(a_i) = \sum_{a_j \in \mathcal{P}_i} P^t(a_i|a_j) Map^t(a_j) \quad (9)$$

where \hat{a}_i^t is the activation when frame f_t is passed through the CNN. Map^t at the desired CNN layer is the cEB -R saliency map for f_t . Computationally, the complexity of cEB -R is on the order of a *single* backward pass. Note that for EB -R, $P_N^t(a_j)$ is used instead of $Map^t(a_j)$ in Eqn. 9. The general framework applied to both video action recognition and captioning is summarized in Algorithm 1. Details of each task are discussed in the following two sections.

4.1. Grounding: Video Action Recognition

In this task, we ground the evidence of a specific action using a model trained on action recognition. The task takes as input a video sequence and the action (\mathcal{A}) to be localized, and outputs spatiotemporal saliency maps for this action in the video. We use the CNN-LSTM implementation of [2] with VGG-16 [22] for our action grounding in video. This encodes the temporal information intrinsically present in the actions we want to localize. The CNN is truncated at the $fc7$ layer such that the $fc7$ features of frames feed into the recurrent unit. We use a single LSTM layer.

Performing cEB -R results in a sequence of saliency maps Map^t for $t = 1, \dots, T$ at $conv5$ (various layers perform similarly [34]). These maps are then used to perform the temporal grounding for action \mathcal{A} . Localizing the action, entails the following sequence of steps. First, the sum of every saliency map is computed to give a vector $\mathcal{S} \in \mathbb{R}^T$. Second, we find an anchor map with the highest sum. Third, we extend a window around the anchor map in both directions in a greedy manner until a saliency map with a negative sum is found. A negative sum indicates that the map is less relevant to the action \mathcal{A} under consideration. This allows us to determine the start and end points of the temporal grounding, $s_{\mathcal{A}}$ and $e_{\mathcal{A}}$ respectively. Fig. 3 depicts the cEB -R pipeline for the task of action grounding.

4.2. Grounding: Video Captioning

In this task, we ground evidence of word(s) using a model trained on video captioning. The task takes as input a video and word(s) to be localized, and outputs spatiotemporal saliency maps corresponding to the query word(s). We use the captioning model of [25] to test our cEB -R approach. This model consists of a VGG-16, followed by a mean pooling of the VGG $fc7$ features, followed by a two-layer LSTM. Fig. 4 depicts cEB -R for caption grounding.

We backpropagate an indicator vector for the words to be visualized starting at the time-steps they were predicted, through time, to the average pooling layer. We then distribute and backpropagate probabilities among frames - according to their forward activations (Eqn. 8) - through the VGG until the $conv5$ layer where we obtain the corresponding saliency map. Performing cEB -R results in a sequence of saliency maps Map^t for $t = 1, \dots, T$ grounding the words in the video frames. Temporal localization is performed using the steps described in Sec. 4.1.

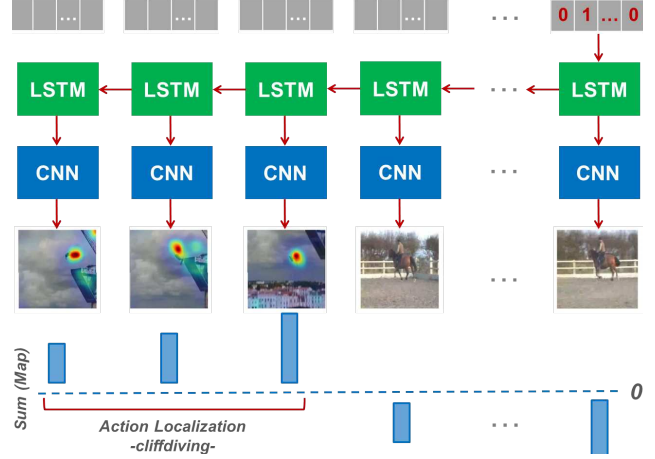


Figure 3: Grounding Action Recognition. The red arrows depict cEB -R for spatiotemporal grounding of the action *CliffDiving*. Starting from the last LSTM time-step, cEB -R backpropagates the probability distribution through time and through the CNN at every time-step. The saliency map for each time-step is used for the spatial localization. The sum of each saliency map, over time, is then used for temporal localization of the action within the video, as described in Sec. 4.1.

Algorithm 1: cEB -R

Input: T -frame video clip, pre-trained CNN-LSTM model, \mathcal{A} : action or word to be localized in the video.

Output: Spatial saliency maps of \mathcal{A} : Map^t for $t = 1, \dots, T$.

Procedure:

- 1 Set a one-hot vector according to the desired action class or caption word \mathcal{A} at the desired n^{th} time-step;
 - 2 Backprop the indicator vector through time and down to the fc CNN layer using EB -R obtaining a saliency map Map^t at every time step t ;
 - 3 Normalize the resulting frame-wise saliency maps over time such that $\sum_{t=1}^T Map^t = 1$;
 - 4 Repeat the above steps, with negated weights at the top layer to get a second set of T saliency maps;
 - 5 Contrastive Operation: Subtract the resulting maps at the fc CNN layer to yield cEB for each time step;
 - 6 Continue EB through the CNN to the desired $conv$ layer to obtain the spatial grounding;
 - 7 The sum of each spatial saliency map over time can be used to perform temporal grounding for \mathcal{A} ;
-

5. Experiments: Action Grounding

In this work we ground the decisions made by our deep models. In order to evaluate this grounding, we compare it with methods that localize actions. Although our framework is able to jointly localize actions in space and time, we report results for spatial localization and temporal localization separately due to the lack of an action dataset that has untrimmed videos with spatiotemporal bounding boxes.

5.1. Spatial Localization

In this section we evaluate how well we ground actions in space. We do this by comparing our grounding results with ground-truth bounding boxes localizing actions per-frame.

Dataset. *THUMOS14* [5] provides per-frame bounding box annotations of humans performing actions for 3207 videos of 24 classes from the *UCF101* dataset [23]. *UCF101* is a trimmed video dataset containing 13320 actions belonging to 101 action classes.

Baselines. We compare our formulation against spatial top-down saliency using a CNN (treating every video frame as an independent image). We also compare against standard backpropagation (BP), and BP for RNNs (BP-R).

Models. We use the following CNN model: VGG-16 of Ma *et al.* [11] trained on *UCF101* video frames and BU101 web images for action recognition with a test accuracy of 83.5%. We use the following CNN-LSTM model: the same VGG-16 fine-tuned with a one-layer LSTM on *UCF101* for action recognition with a test accuracy of 83.3%.

Setup and Results. We use the bounding box annotations to evaluate our spatial grounding using the pointing game introduced by Zhang *et al.* [34]. We locate the point having maximum value on each top-down saliency map. Following [34], if a 15-pixel diameter circle around the located point intersects the ground-truth bounding-box of the action category for a frame, we record a hit, otherwise we record a miss. We measure the spatial action localization accuracy by $Acc = \#Hits / (\#Hits + \#Misses)$ over all the annotated frames for each action.

Table 1 reports the results of the spatial pointing game. Extending top-down saliency in time (-R) consistently improves the accuracy for all three methods, compared to performing top-down saliency separately on every frame of the video using a CNN. EB-R has the greatest absolute improvement of 5.7%.

We note that the non-contrastive versions outperform their contrastive counterparts. This is because they highlight discriminative evidence for actions, which may not necessarily be the humans performing the actions. For example, for many actions in *UCF101*, the human may be in a standing position, in which case *cEB-R* will highlight cues that are discriminative and unique to this action rather than highlighting the human. These cues may belong to the context in which the activity is performed, or the action classes

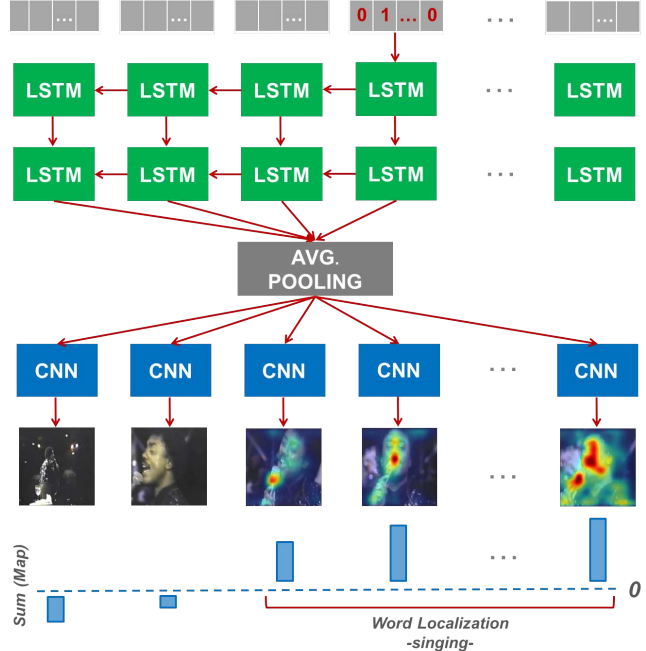


Figure 4: Grounding Captioning. The red arrows depict *cEB-R* for spatiotemporal caption grounding. The video caption produced by the model is *A man is singing on a stage*. Starting from the time-step corresponding to the word *singing*, *cEB-R* backprops the probability distribution through the previous time-steps and through the CNN. The saliency map for each time step is used for spatial localization. The sum of each saliency map, over time, is then used for temporal localization of the word within the clip.

Method Acc (%)					
EB	EB-R	<i>cEB</i>	<i>cEB-R</i>	BP	BP-R
55.8	61.5	37.0	39.1	37.3	39.2

Table 1: Accuracy of the spatial pointing game conducted on $\sim 3K$ videos of *UCF101* for spatially locating humans performing actions in videos. The results show that extending top-down saliency in time (-R) improves the accuracy compared to performing top-down saliency separately on every frame of the video using a CNN. The non-contrastive versions work better for reasons described in the text.

on which the model was trained. We demonstrate this in Fig. 5 for the actions *Surfing* and *BasketballDunk*.

5.2. Temporal Localization

In this section we evaluate how well we ground actions in time. We do this by comparing our grounding results with ground-truth action boundaries.

Datasets. We first use a simple and controlled setting to validate our method by creating a synthetic action detection dataset. We then present results on the *THUMOS14* [5] action detection dataset. The synthetic dataset is created by concatenating two *UCF101* videos uniformly sampled: a ground truth (*GT*) video, and a random (*rand*) back-



(a) Grounding *Surfing* using EB-R (L) and cEB-R (R)



(b) Grounding *BasketballDunk* using EB-R (L) and cEB-R (R)

Figure 5: The saliency maps produced by EB-R (left) and cEB-R (right) together with the *THUMOS14* groundtruth bounding box over the same frame of the actions (a) *Surfing* and (b) *BasketballDunk*. In both cases, EB-R highlights the most salient regions of the frame for this action (human), which matches the bounding box annotation. However, cEB-R highlights the region that is unique to the ground truth action: the waves for *Surfing*, and the hoop for *BasketballDunk*. This is because highlighting the human region does not provide insightful information to the classifier.

ground video, such that $\text{class}(GT) \neq \text{class}(rand)$. The two actions are concatenated, first sequentially ($rand + GT$ or $GT + rand$) in 16-frame clips, then inserted at a random position ($rand + GT + rand$) in 128-frame clips. We use all 3783 test videos provided in *UCF101*, each in combination with a different random background video. The *THUMOS14* dataset consists of 1010 untrimmed validation videos and 1574 untrimmed test videos of 20 action classes. Among test videos, we evaluate our framework on the 213 test videos which contain annotations as in [27, 19].

Baselines. For the synthetic experiment, we compare cEB-R and EB-R with a probability-based approach where we threshold the predicted probability (to 1 if ≥ 0.5 , to -1 if < 0.5) of the *GT* class at every time-step. For the detection experiment in *THUMOS14* we compare our proposed method with state-of-the-art approaches.

Models. For the synthetic dataset, we use the same CNN-LSTM model used for spatial action grounding (Sec. 5.1). For the *THUMOS14* dataset we use a CNN-LSTM model: the same VGG-16 model used for spatial action grounding (Sec. 5.1) fine-tuned with a one-layer LSTM on *UCF101* and trimmed sequences from *THUMOS14* background and validation sets.

Setup and Results: Synthetic Data. First, we perform experiments on the synthetic videos composed of two sequential actions, where the boundary is the midpoint. Fig. 6 presents a sample spatiotemporal localization. The heatmaps produced by cEB-R correctly ground the queried action. More examples are presented in the supplementary material. While Fig. 6 presents a qualitative sample, Fig. 7 quantitatively presents results on the entire test set. The ac-



(a) Grounding of the action *TableTennisShot* in the video



(b) Grounding of the action *Skiing* in the video

Figure 6: Applying *contrastive* Excitation Backprop for Recurrent Networks (cEB-R) to produce spatiotemporal localization of actions in sample frames of a video. Demonstrated here is (a) cEB-R spatiotemporal localization of *TableTennisShot* in a video (b) cEB-R spatiotemporal localization of *Skiing* in the same video. The video consists of two consecutive actions that are synthetically concatenated: *Skiing* followed by *TableTennisShot*.

tion switches from *GT* to *rand* or vice versa midway. It can be seen that the sum of saliency maps is: positive and increasing as more of the *GT* action is observed, negative and decreasing as more of the *rand* action is observed.

Next, we perform experiments where we vary the length of the *GT* action that we want to localize inside a clip. To retain action dynamics, we sample *GT* and *rand* from the entire length of their corresponding videos. Table 2 presents the temporal localization results of our synthetic data. In the experimental setup with fixed action length we assume that we know the label and length of the action to be localized. To localize, we find the highest consecutive sum of attention maps for the desired action length. Regarding the sequences with unknown action lengths, we only assume the label of the action to be localized and perform the pipeline described in Sec. 4.1. In the bottom half of Table 2 we only report thresholded probabilities and cEB-R results since our localization procedure assumes negative values at action boundaries, whereas EB-R is non-negative. The grounded evidence obtained by cEB-R attains the highest detection scores, 73.5% and 62.0%, for action sequences of known and unknown lengths, respectively, for IoU overlap between detections and ground-truth of $\alpha = 0.5$, despite the fact that the model is not trained for localization.

Setup and Results: THUMOS14 Pointing Game. We evaluate the pointing game in time for *THUMOS14* -a fair evaluation for methods that do not optimize for detection. For processing, we divide a video into 128-frame consecutive clips. We perform the pointing game by pointing [34] in time to the peak sum of saliency maps. For each ground-truth annotation we check if the detected peak is within its boundaries. If yes, we count it as a hit, otherwise, as a miss. We compare this approach with the peak position of pre-

dicted probabilities, and a random point in that clip.

The results of this experiment are presented in Table 3. Pointing to a random position clearly obtains lowest results while peak probability (65.8%) and *cEB-R* (65.1%) have similar performance. However, peak probability does not offer spatial localization. Peak probability uses the model prediction, while *cEB-R* uses the evidence of that prediction. Moreover, we observe that peak probability and *cEB-R* are complementary, yielding 77.4%.

Setup and Results: THUMOS14 Action Detection.

We evaluate how well our grounding does on the more challenging task of action detection that it was not trained for. In this experiment, we divide a video into 128-frame consecutive clips for processing. Table 4 presents the temporal detection results of the THUMOS14 dataset. Differently from the pointing game experiment, we detect the start and end of the ground-truth action. We note that although our method is not supervised for the detection task, we achieve an accuracy of 57.9% when locating a ground truth class with an overlap $\alpha = 0.1$ as demonstrated in Table 4.

6. Experiments: Caption Grounding

In this section, we show how *cEB-R* is also applicable in the context of caption grounding. As observed by [15], there is an absence of datasets with spatiotemporal annotations of frames for captions. Therefore, they propose the following experimental setup which we follow: qualitative results for the spatiotemporal grounding on videos, and quantitative results for spatial grounding on images.

Datasets. We use the MSR-VTT [28] dataset for video captioning and Flickr30kEntities [14] for image captioning.

Models. We use the CNN-LSTM-LSTM video captioning model of [25] trained on MSR-VTT to test our *cEB-R* approach for spatiotemporal grounding as described in Sec. 4.2. We use the same video captioning model, without the average pooling layer, trained on Flickr30kEntities for image captioning. The models have comparable METEOR scores to the Caption-Guided Saliency work of [15], to which we compare our results: 26.5 (vs. 25.9) for video captioning and 18.0 (vs. 18.3) for image captioning.

Setup and Results. For the MSR-VTT video dataset, we sample 26 frames per video following [15] and perform grounding of nouns. Fig. 8 presents the grounding for the word *man* and *phone* in the same video. The *man* is well localized only in frames where a man appears, and the *phone* is well localized in frames where a phone appears.

We quantitatively evaluate our results of spatial grounding using the pointing game on the Flickr30kEntities and compare our method to the Caption-Guided Saliency work of [15], following their evaluation protocol. We use ground truth captions as an input to our model in order to reproduce the same captions. Then, we use bounding box annotations for each noun phrase in the ground truth captions and check

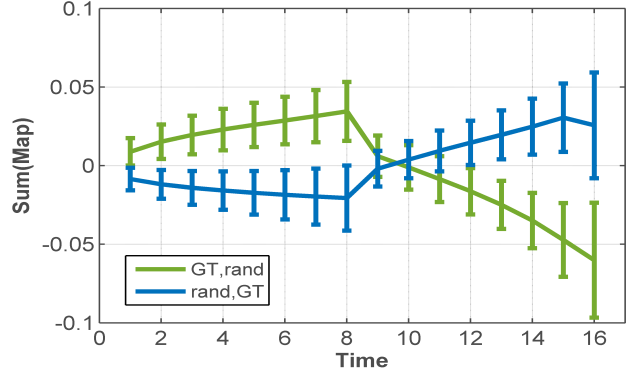


Figure 7: Sum of the saliency maps at *fc7* over time, in frames, for synthetic videos that (blue) have a *rand* action followed by a *GT* action and (green) have a *GT* followed by a *rand* action. The average and standard deviation are reported over all test videos. *cEB-R* provides an accurate midway boundary between actions.

		Length (frames)	Probability (%)	EB-R (%)	<i>cEB-R</i> (%)
Action Length	Known	11	8.5	11.3	15.5
		41	28.2	38.5	53.2
		65	47.7	56.3	73.5
	Unknown	11	3.4	-	4.1
		41	9.5	-	47.9
		65	35.7	-	62.0

Table 2: Action detection results on synthetic data, measured by mAP at IoU threshold $\alpha = 0.5$. Top part of table: methods assume that the length and label of the action to be detected are known. Bottom part of table: methods assume that the label is known, but the length is unknown. *cEB-R* attains best performance.

Method	Accuracy (%)
Random	57.3
Peak probability	65.8
<i>cEB-R</i>	65.1
Peak probability + <i>cEB-R</i>	77.4

Table 3: Pointing game in time performed on the THUMOS14 test set. The probability of an action together with the evidence for presence of the action are complementary and give a great improvement in accuracy when combined.

whether the maximum point in a saliency map is inside the annotated bounding box.

Table 5 shows the results of the spatial pointing game on Flickr30kEntities. Our approach achieves comparable performance to [15]. In this experiment, we ground the ground truth captions to match the experimental setup in [15]. Although we follow their protocol for fair comparison, we note that our method can better highlight evidence using generated captions (vs. ground truth captions). This is because the evidence of a ground truth noun that is not

Method	mAP ($\alpha = 0.1$)
Karaman <i>et al.</i> [7]	4.6
Wang <i>et al.</i> [26]	18.2
Oneata <i>et al.</i> [12]	36.6
Richard <i>et al.</i> [17]	39.7
Shou <i>et al.</i> [20]	47.7
Yeung <i>et al.</i> [31]	48.9
Yuan <i>et al.</i> [32]	51.4
Xu <i>et al.</i> [27]	54.5
Zhao <i>et al.</i> [35]	60.3
Kaufman <i>et al.</i> [9]	61.1
Ours	57.9

Table 4: Our approach vs. fully supervised approaches for action detection on *THUMOS14*, measured by mAP at IoU threshold $\alpha = 0.1$. Although our model is not trained for action detection (trained for recognition), we achieve 57.9%, which is comparable to state-of-the-art when localizing a ground truth action in a video.

Method	Avg (Noun Phrases)
Baseline random	0.268
Baseline center	0.492
Caption-Guided Saliency [15]	0.501
Ours	0.512

Table 5: Evaluation of spatial saliency on *Flicker30kEntities* using *cEB-R*. Baseline random samples the maximum point uniformly and Baseline center always picks the center.

predicted may not be sufficiently activated in the forward pass. Fig. 9 presents some visual examples of grounding in images using the generated captions.

Our approach has a computational advantage over [15]. In order to obtain spatial saliency maps for a word in a video, *cEB-R* requires one forward pass and one backward pass through the CNN-LSTM-LSTM, while [15] requires one forward pass through the CNN part, but m forward passes through the LSTM-LSTM part, where $m = 64$ is the area of the saliency map (vs. our single backward pass). Moreover, they require f forward LSTM passes, where $f = 26$ is the number of frames, to compute the temporal grounding, whereas ours is implicitly spatiotemporal.

7. Conclusion

In conclusion, we devise a temporal formulation, *cEB-R*, that enables us to visualize how recurrent networks ground their decisions in visual content. We apply this to two video understanding tasks: video action recognition, and video captioning. We demonstrate how spatiotemporal top-down saliency is capable of grounding evidence on several action and captioning datasets. These datasets provide annotations for detection and/or localization, to which we have compared the evidence in our generated saliency maps. We observe the strengths of *cEB-R* in highlighting discrimina-

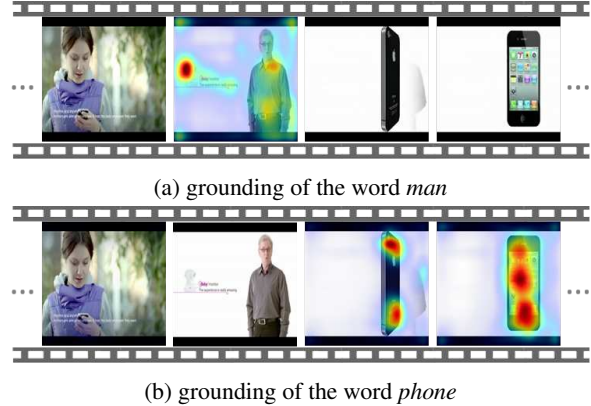


Figure 8: Comparison of grounding of words *man* and *phone* in the caption *A man is talking about a phone* of a video from *MSR-VTT* using *cEB-R*. The man is well localized in (a) and the phone is well localized in (b), as desired.

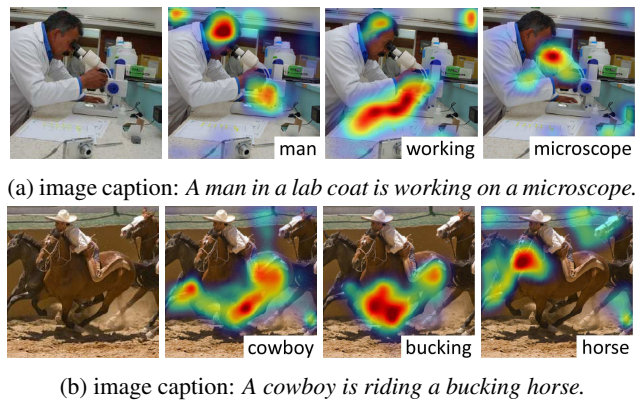


Figure 9: Grounding different words of a caption using *cEB-R* for two images from the *Flicker30kEntities* dataset.

tive evidence, which was particularly beneficial for temporal grounding. We also observe the strengths of its variant, *EB-R*, in highlighting salient evidence, which was particularly beneficial for spatial localization of action subjects.

Acknowledgments

We thank Kate Saenko and Vasili Ramanishka for helpful discussions. This work was supported in part by NSF grants 1551572 and 1029430, an IBM PhD Fellowship, gifts from Adobe and NVidia, and Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00341. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

- [1] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *IEEE International Conference on Computer Vision*, pages 2956–2964, 2015. 1
- [2] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015. 4
- [3] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE international conference on computer vision (ICCV)*, volume 00, pages 3449–3457, Oct. 2018. 2
- [4] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 2016. 3
- [5] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. <http://csrcv.ucf.edu/THUMOS14/>, 2014. 5
- [6] R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *32nd International Conference on Machine Learning (ICML)*, pages 2342–2350, 2015. 3
- [7] S. Karaman, L. Seidenari, and A. Del Bimbo. Fast saliency based pooling of Fisher encoded dense trajectories. In *ECCV THUMOS Workshop*, volume 1, page 5, 2014. 8
- [8] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. In *ICLR Workshop*, 2016. 1, 2
- [9] D. Kaufman, G. Levi, T. Hassner, and L. Wolf. Temporal tessellation: A unified approach for video analysis. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 8
- [10] J. Li, X. Chen, E. Hovy, and D. Jurafsky. Visualizing and understanding neural models in nlp. In *NAACL-HLT*, pages 681–691, 2016. 2
- [11] S. Ma, S. A. Bargal, J. Zhang, L. Sigal, and S. Sclaroff. Do less and achieve more: Training cnns for action recognition utilizing action images from the web. *Pattern Recognition*, 2017. 5
- [12] D. Oneata, J. Verbeek, and C. Schmid. The lear submission at thumos 2014. 2014. 8
- [13] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2
- [14] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2641–2649, 2015. 7
- [15] V. Ramanishka, A. Das, J. Zhang, and K. Saenko. Top-down visual saliency guided by captions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 7, 8
- [16] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016. 2
- [17] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3131–3140, 2016. 8
- [18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 618–626, 2017. 1, 2
- [19] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. *arXiv preprint arXiv:1703.01515*, 2017. 6
- [20] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058, 2016. 8
- [21] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1, 2
- [22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4
- [23] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5
- [24] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014. 1, 2
- [25] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *North American Chapter of the Association for Computational Linguistics Human Language Technologies NAACL-HLT*, 2015. 4, 7
- [26] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 1(2):2, 2014. 8
- [27] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. 2017. 6, 8
- [28] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016. 7
- [29] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 2

- [30] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4507–4515, 2015. [2](#)
- [31] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2678–2687, 2016. [8](#)
- [32] J. Yuan, B. Ni, X. Yang, and A. A. Kassim. Temporal action localization with pyramid of score distribution features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3093–3102, 2016. [8](#)
- [33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833. Springer, 2014. [1](#), [2](#)
- [34] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision (ECCV)*, pages 543–559. Springer, 2016. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#)
- [35] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. [8](#)
- [36] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016. [1](#), [2](#)