# MARGINALIZED AVERAGE ATTENTIONAL NETWORK FOR WEAKLY-SUPERVISED LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In weakly-supervised temporal action localization, previous works suffer from overestimating the most salient regions and fail to locate dense and integral regions for each entire action. To alleviate this issue, we propose a marginalized average attentional network (MAAN) to suppress the dominant response of the most salient regions in a principled manner. The MAAN employs a novel marginalized average aggregation (MAA) module and learns a set of latent discriminative probabilities in an end-to-end fashion. MAA samples the subsets from the video snippet features based on the latent discriminative probabilities and takes the expectation over all the subset features. Theoretically, we prove that the learned latent discriminative probabilities reduce the difference of responses between the most salient regions and the others, and thus MAAN generates better class activation sequences to identify more dense and integral action regions in the videos. Moreover, we propose a fast algorithm to reduce the complexity of constructing MAA from $O(2^T)$ to $O(T^2)$. Extensive experiments on two large-scale video datasets show that our MAAN achieves superior performance on weakly-supervised temporal action localization task.

## 1 INTRODUCTION

The recent weakly-supervised setting for temporal action localization interested to the community is to train the model with solely the video-level class labels which are more easily collected, and to predict both the action class and the temporal boundary of each action instance at the test time. The main difficulty in solving the weakly-supervised localization task is how to find the right way to express and infer the underlying location information with only the video-level class labels. Traditionally, this is achieved by explicitly densely sampling several possible instances at various locations with various durations (Bilen & Vedaldi, 2016; Kantorov et al., 2016; Zhang et al., 2017), and then training instance-level classifiers with multiple instances learning (Cinbis et al., 2017; Yuan et al., 2017a) or curriculum learning (Bengio et al., 2009). However, the length of action and video varies too much such that the number of instance proposals varies a lot and can be huge.

Recent research, however, has pivoted to acquire the location information by generating the class activation sequence (CAS) directly (Nguyen et al., 2017), which produces the classification score sequence of being each action for each snippet over time. The CAS along the 1D temporal dimension for a video is inspired by the class activation map (CAM) (Zhou et al., 2016; 2014; Pinheiro & Collobert, 2015; Oquab et al., 2015) in weakly-supervised object detection. The CAM-based models have shown that despite being trained on image-level labels, convolutional neural networks (CNNs) have the remarkable ability to localize objects. Similar to the object detection, the basic idea behind the CAS-based methods for action localization in the training is to sample the non-overlapped snippets from a video, then to aggregate the snippet-level features into a video-level feature, and finally to yield a video-level class prediction. During testing, the model generates a CAS for each class that identifies the discriminative action regions, and a threshold is then applied on the CAS to localize each action instance in terms of the start time and the end time.

In CAS-based methods, the feature aggregator that aggregates several snippet-level features into a video-level feature is the critical building block of weakly-supervised neural networks. A model's ability to capture the location information of action is primarily determined by the design of aggregators. While using the global average pooling over a full image or across the video snippets has shown

great promise of identifying the discriminative regions (Zhou et al., 2016; 2014; Pinheiro & Collobert, 2015; Oquab et al., 2015), treating each pixel or snippet equally loses the opportunity to benefit from several more essential parts. Some recent works (Nguyen et al., 2017; Zhu et al., 2017) have tried to learn attentional weights for different snippets to compute a weighted sum as the aggregated feature. However, it suffers the problem of easily dominated by only a few most salient snippets.

In general, models trained with only video-level class labels are easily responsive to small and sparse discriminative regions from the snippets of interest, which deviates from the requirement of the localization task that to locate dense and integral regions for each entire action. To mitigate this gap and reduce the effect of the domination by the most salient regions, several works propose additional heuristic tricks to the previous models. For example, (Wei et al., 2017; Zhang et al., 2018b) attempt to heuristically erase the current mined most salient regions predicted by the model, and force the network to attend other salient regions in the remaining regions by forwarding the model several times. However, the heuristic multiple-run model is not end-to-end trainable. Moreover, it is the ensemble of multiple-run mined regions but not the single model's own ability that learns the entire action regions. "Hide-and-seek"(Singh & Lee, 2017) randomly masks out some regions of the input during training, enforcing the model to localize other salient regions when the most salient regions happened to be masked out. However, due to the uniform prior, all the input regions are masked out with the same probability. It is very likely that most of the time it is the background that being masked out.

To this end, we propose the marginalized average attentional network (MAAN) to alleviate the effect of the domination by the most salient region in an end-to-end fashion for weakly-supervised action localization. Specifically, MAAN suppresses the action prediction response of the most salient regions by employing marginalized average aggregation (MAA) and learning the latent discriminative probability in a principled manner. Unlike the previous attentional pooling aggregator, which calculates the weighted sum with attention weights, MAA first samples a subset of features according to their latent discriminative probability, and then calculates the average of these sampled features. Finally, MAA takes the expectation (marginalization) of the average aggregated subset features over all the possible subsets to achieve the final aggregation. It not only alleviates the dominant issues of the most salient regions, but also keeps the scale of the aggregated feature in a stable range. We theoretically prove that, with the MAA, the learned latent discriminative probability indeed reduces the difference of response between the most salient regions and the others. Therefore, MAAN can identify more dense and integral regions for each action. Moreover, since enumerating all the possible subsets is exponential expensive, we further propose a fast iterative algorithm to reduce the complexity of the expectation calculation and provide the proof of it theoretically. Furthermore, since all the components of the network are differentiable, MAAN is easy to be trained in an end-to-end fashion. Extensive experiments on two large-scale video datasets show that MAAN consistently outperforms the baseline models and achieves superior performance on weakly-supervised temporal action localization task.

In summary, our main contributions include: (1) a new end-to-end learnable marginalized average attentional network (MAAN) with a marginalized average aggregation (MAA) module in the weakly-supervised setting; (2) theoretical analysis of the properties of MAA and the reasons why MAAN alleviates the dominant issues of the most salient regions in weakly-supervised action localization; (3) a fast iterative algorithm to effectively reduce the computational complexity of the MAA; and (4) the superior performance on two benchmark video datasets, THUMOS14 and ActivityNet1.3, on the weakly-supervised temporal action localization task.

## 2 Marginalized Average Attentional Network

In this section, we describe our marginalized average attentional network (MAAN) for weakly-supervised temporal action localization. We first derive the formulation of the feature aggregation module in MAAN as a marginalized average aggregation (MAA) procedure in Sec. 2.1. Then, we analyze the properties of MAA in Sec. 2.2, and present our recurrent fast computation algorithm for MAA construction in Sec. 2.3. Finally, we describe our network architecture that incorporates this feature aggregation module, and introduce the inference with it on weakly-supervised temporal action localization task in Sec. 2.4.

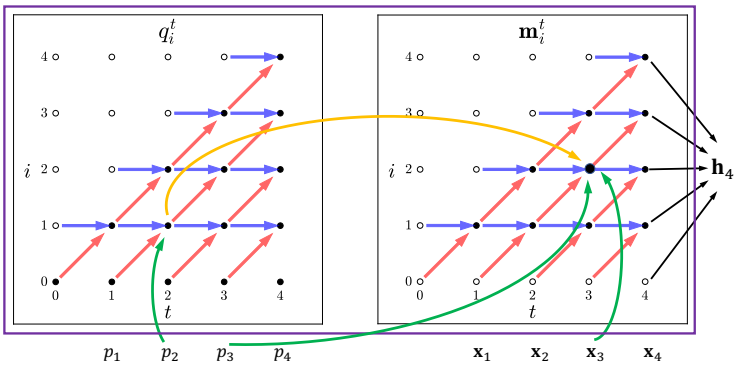


Figure 1: The purple box demonstrates the marginalized average aggregation module, where the inputs are $\{p_i\}_{i=1}^4$ and $\{\mathbf{x_i}\}_{\mathbf{i=1}}^{\mathbf{4}}$ and the output is $\mathbf{h_4}$. The two black boxes are the demonstration of computation graph of $q_i^t$ and $\mathbf{m_i^t}$, respectively. The black hollow point indicates its value is 0, while the value of the black solid point is non-zero. $q_0^0$ is initialized as 1.

### 2.1 Marginalized Average Aggregation

Let us denote the snippet-level features to be aggregated as a set of representations $\{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T\}$, where $\mathbf{x}_t \in \mathbb{R}^m$ is the $m$ dimensional feature representation extracted from a video snippet centered at time $t$ and $T$ is the total number of sampled video snippets. The conventional attentional weighted sum pooling will aggregate the input snippet-level features into a video-level representation $\overline{\mathbf{x}}$. Denote the set of attentional weights corresponding to the snippet-level features as $\{\lambda_1, \lambda_2, \cdots \lambda_T\}$, where $\lambda_t$ is a scalar attentional weight for $\mathbf{x}_t$. Then the aggregated video-level representation is calculated as

$$\overline{\mathbf{x}} = \sum_{t=1}^{T} \lambda_t \mathbf{x}_t. \tag{1}$$

Different from the conventional aggregation mechanism, the proposed MAA module aggregates the features by firstly generating a set of binary indicators to determinate whether a snippet to be sampled or not. We then calculate the average aggregation of these sampled snippet-level representations. Lastly, we calculate the expectation (marginalization) of the aggregated average feature for all the possible subsets, and obtain the proposed marginalized average aggregated feature. Formally, in our proposed marginalized average aggregation module, we first denote a set of probabilities $\{p_1, p_2, \cdots p_T\}$, where each $p_t \in [0, 1]$ is a scalar corresponding to $\mathbf{x_t}$, similar to the $\lambda_t$. We then sample a set of random variables $\{z_1, z_2, \cdots z_T\}$, where $z_t \sim Bernoulli(p_t)$, i.e., $z_t \in \{0, 1\}$ with probability $P(z_t = 1) = p_t$. The sampled set is used to represent the subset selection of snippet-level features, in which $z_t = 1$ indicates $\mathbf{x}_t$ is selected, otherwise not. Therefore, the average aggregation of the sampled subset snipped-level representations is given by $\overline{\mathbf{s}} = \sum_{i=1}^{T} z_i \mathbf{x}_i / \sum_{i=1}^{T} z_i$ , and our proposed aggregated feature, defined as the expectation of all the possible subset-level average aggregated representations, is given by

$$\overline{\mathbf{x}} = \mathbb{E}[\overline{\mathbf{s}}] = \mathbb{E}\left[\frac{\sum_{i=1}^{T} z_i \mathbf{x}_i}{\sum_{i=1}^{T} z_i}\right]. \tag{2}$$

### 2.2 Partial order Preservation and Dominant Response Suppression

Direct learning and prediction with the attention weights $\lambda$ in Eq.(1) lean to over response to the most salient region in weakly-supervised action localization. The MAA in Eq.(2) has two properties that are useful to alleviate the domination effect of the most salient regions. First, partial order preservation property, i.e., the latent discriminative probabilities preserve the partial order with respect to their attention weights. Second, dominant response suppression property, i.e., the differences of the latent discriminative probabilities between the most salient items and others are smaller than the differences

between their attention weights. The partial order preservation property guarantees that it does not mix up the action and non-action snippets by assigning a high latent discriminative probability to a snippet with low response. Dominant response suppression property enables to reduce the dominant effect of the most salient regions and encourages to identify dense and more integral action regions. Formally, we present the two properties in Proposition 1 and Proposition 2, respectively. Detail proofs can be found in the Appendix.

**Proposition 1.** *Let $z_i \sim Bernoulli(p_i)$ for $i \in \{1, ..., T\}$. Then for $T \geq 2$, Eq.(3) holds true. And $p_i \geq p_j \Leftrightarrow c_i \geq c_j \Leftrightarrow \lambda_i \geq \lambda_j$.*

$$\mathbb{E}\left[\frac{\sum_{i=1}^{T} z_i \mathbf{x}_i}{\sum_{i=1}^{T} z_i}\right] = \sum\nolimits_{i=1}^{T} c_i p_i \mathbf{x}_i = \sum\nolimits_{i=1}^{T} \lambda_i \mathbf{x}_i \tag{3}$$

*where $c_i = \mathbb{E}\left[1/(1 + \sum_{k=1,k\neq i}^{T} z_k)\right]$ and $c_j = \mathbb{E}\left[1/(1 + \sum_{k=1,k\neq j}^{T} z_k)\right]$ and $\lambda_i = c_i p_i$*

Proposition 1 shows that the latent discriminative probabilities $\{p_i\}$ preserve the partial order of the attention weights $\{\lambda_i\}$. It means that a large attention weight corresponds to a large discriminative probability. Thus, it guarantees that the latent discriminative probabilities preserve the ranking of action prediction response. Eq. (3) can be seen as a factorization of the attention weight $\lambda_i$ into the multiplication of two components, $p_i$ and $c_i$, for $i \in \{1, ..., T\}$. $p_i$ is the latent discriminative probability related to the feature of snippet $i$ itself. The factor $c_i$ captures the contextual information of snippet $i$ from the other snippets. This factorization can be considered as introducing structure information into the aggregation. The factor $c_i$ can be considered as performing a structural regularization for learning the latent discriminative probabilities $p_i$ for $i \in \{1, ..., T\}$, as well as for learning the more informative aggregation.

**Proposition 2.** *Let $z_i \sim Bernoulli(p_i)$ for $i \in \{1, ..., T\}$ . Denote $c_i = \mathbb{E}\left[1/(1 + \sum_{k=1,k\neq i}^{T} z_k)\right]$ and $\lambda_i = c_i p_i$ for $i \in \{1, ..., T\}$. Denote $\mathcal{I} = \left\{i \,\middle|\, c_i \geq 1/(\sum_{t=1}^{T} p_t)\right\}$ as an index set. Then $\mathcal{I} \neq \emptyset$ and for $\forall i \in \mathcal{I}$, $\forall j \in \{1, ..., T\}$ inequality (4) holds true*

$$\left|\frac{p_i}{\sum_{t=1}^{T} p_t} - \frac{p_j}{\sum_{t=1}^{T} p_t}\right| \leq \left|\frac{\lambda_i}{\sum_{t=1}^{T} \lambda_t} - \frac{\lambda_j}{\sum_{t=1}^{T} \lambda_t}\right| \tag{4}$$

The index set $\mathcal{I}$ can be viewed as the most salient feature set. Proposition 2 shows that the difference between the normalized latent discriminative probabilities of the most salient regions and others is smaller than the difference between their attention weights. It means that the prediction for each snippet with the latent discriminative probability can reduce the gap between the most salient feature and others compared with employing attention weights. Thus, MAAN suppresses dominant responses of the most salient feature and encourages to identify dense and more integral action regions.

Directly learning the attention weights $\lambda$ leans to over response to the most salient region in weakly-supervised temporal localization. Namely, attention weights for only a few snippets are too large and dominate the others, while attention weights for most of the other snippets that also belong to the true action are underestimated. Proposition 2 shows that latent discriminative probability is able to reduce the gap between the most salient feature and others compared with attention weights. Thus, by employing the latent discriminative probabilities for prediction instead of the attention weights, our method can alleviate the dominant effect of the most salient region in weakly-supervised temporal localization.

### 2.3 Recurrent Fast Computation

Given a video containing $T$ snippet-level representations, there are $2^T$ possible configurations for the subset selection. Directly summing up all the $2^T$ configurations in the expectation to calculate $\overline{\mathbf{x}}$ has $O(2^T)$ complexity. In order to reduce the exponential complexity, we propose an iterative method to calculate $\overline{\mathbf{x}}$ recurrently with $O(T^2)$ complexity. Let us denote the aggregated feature of $\{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_t\}$ with length $t$ as $\mathbf{h}_t$, and denote $\mathbf{Y_t} = \sum\limits_{i=1}^{t} z_i \mathbf{x}_i$ and $Z_t = \sum\limits_{i=1}^{t} z_i$ for simplicity, then

we have a set of

$$\mathbf{h}_t = \mathbb{E}\left[\frac{\sum_{i=1}^{t} z_i \mathbf{x}_i}{\sum_{i=1}^{t} z_i}\right] = \mathbb{E}\left[\frac{\mathbf{Y_t}}{Z_t}\right], t \in \{1, 2, \cdots, T\}, \tag{5}$$

and the aggregated feature of $\{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T\}$ can be obtained as $\overline{\mathbf{x}} = \mathbf{h}_T$. In Eq.(5), $Z_t$ is the summation of all the $z_i$, which indicates the number of elements selected in the subset. Although there are $2^t$ distinct configurations for $\{z_1, z_2, \cdots z_t\}$, it is only $t+1$ distinct values for $Z_t$, i.e. $0, 1, \cdots, t$. Therefore, we can divide all the $2^t$ distinct configurations into $t+1$ groups, where the configurations sharing with the same $Z_t$ fall into the same group. Then the expectation $\mathbf{h}_t$ can be calculated as the summation of the $t+1$ parts. That is, $\mathbf{h}_t = \mathbb{E}\left[\mathbb{E}\left[\frac{\mathbf{Y_t}}{Z_t}\middle| Z_t = i\right]\right] = \sum_{i=0}^{t} \mathbf{m}_i^t$, where the $\mathbf{m}_i^t$, indicating the $i^{th}$ part of $\mathbf{h}_t$ for group $Z_t = i$, is shown in Eq.(6).

$$\mathbf{m}_i^t = P\left(Z_t = i\right) \mathbb{E}\left[\frac{\mathbf{Y_t}}{Z_t}\middle| Z_t = i\right]. \tag{6}$$

In order to calculate $\mathbf{h}_{t+1} = \sum_{i=0}^{t+1} \mathbf{m}_i^{t+1}$, given $\mathbf{m}_i^t$ , $i \in \{0, \cdots, t\}$, we can calculate $\mathbf{m}_i^{t+1}$, $i \in \{0, 1, \cdots, t+1\}$ recurrently. The key idea here is that $\mathbf{m}_i^{t+1}$ comes from two cases: if $z_{t+1} = 0$, then $\mathbf{m}_i^{t+1}$ is the same as $\mathbf{m}_i^t$; if $z_{t+1} = 1$, then $\mathbf{m}_i^{t+1}$ is the weighted average of $\mathbf{m}_{i-1}^t$ and $\mathbf{x}_{t+1}$. The latter case is also related to the probability $P\left(Z_t = i-1\right)$. By denoting $q_{i-1}^t = P\left(Z_t = i-1\right)$ for simplicity, we can obtain $\mathbf{m}_i^{t+1}$ as a function of several elements:

$$\mathbf{m}_i^{t+1} = f(\mathbf{m}_{i-1}^t, \mathbf{m}_i^t, \mathbf{x}_{t+1}, p_{t+1}, q_{i-1}^t). \tag{7}$$

Similarly, the computation of $q_i^{t+1} = P\left(Z_{t+1} = i\right)$ comes from two cases: the probability of selecting $i-1$ items from the first $t$ items and selecting the $(t+1)^{th}$ item, i.e., $q_{i-1}^t p_{t+1}$; and the probability of selecting $i$ items all from the first $t$ items and not selecting the $(t+1)^{th}$ item, i.e., $q_i^t\left(1 - p_{t+1}\right)$. We derive the function of $\mathbf{m}_i^{t+1}$ and $q_i^{t+1}$ in Proposition 3. Detail proofs can be found in the Appendix.

**Proposition 3.** *Let $z_t \sim Bernoulli(p_t)$ , $Z_t = \sum_{i=1}^{t} z_i$ and $\mathbf{Y_t} = \sum_{i=1}^{t} z_i \mathbf{x}_i$ for $t \in \{1, ..., T\}$. Define $\mathbf{m}_i^t$ , $i \in \{0, \cdots, t\}$ as Eq. (6) and $q_i^t = P\left(Z_t = i\right)$, then $\mathbf{m}_i^{t+1}$ $i \in \{0, 1, \cdots, t+1\}$ can be obtained recurrently by Eq.(8) and Eq.(9).*

$$\mathbf{m}_i^{t+1} = p_{t+1}\left(b_{i-1}\mathbf{m}_{i-1}^t + (1 - b_{i-1}) q_{i-1}^t \mathbf{x}_{t+1}\right) + (1 - p_{t+1})\mathbf{m}_i^t, \tag{8}$$

$$q_i^{t+1} = p_{t+1} q_{i-1}^t + \left(1 - p_{t+1}\right) q_i^t, \tag{9}$$

*where $b_i = \frac{i}{i+1}$, $q_{-1}^t = 0$, $q_{t+1}^t = 0$, $q_0^0 = 1$, $\mathbf{m}_0^t = \mathbf{0}$, and $\mathbf{m}_{t+1}^t = \mathbf{0}$.*

Proposition 3 provides recurrent formula to calculate $\mathbf{m}_i^t$. With this recurrent formula, we can calculate the aggregation $\mathbf{h}_T$ by iteratively calculating $\mathbf{m}_i^t$ from $i = 1$ to $t$ and $t = 1$ to $T$. Therefore, we can obtain the aggregated feature of $\{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T\}$ as $\overline{\mathbf{x}} = \mathbf{h}_T = \sum_{i=0}^{T} \mathbf{m}_i^T$. The iterative computation procedure is summarized in Algorithm 1 in the Appendix, where the time complexity is $O(T^2)$.

With the fast iterative algorithm in Algorithm 1, the MAA becomes practical for end-to-end training. A demonstration of the computation graph for $q_i^{t+1}$ in Eq.(9) and $\mathbf{m}_i^{t+1}$ in Eq.(8) is presented in the left and right of Figure. 1, respectively. From the Figure. 1, we can see it clearly that, to compute $\mathbf{m}_2^3$ (the big black node in the right), it needs $\mathbf{m}_1^2$, $\mathbf{m}_2^2$, $\mathbf{x_3}$, $p_3$, and $q_1^2$. The MAA can be easily implemented as a subnetwork for end-to-end training and can be used to replace the operation of other feature aggregators.

### 2.4 Network Architecture and Temporal Action Localization

**Network Architecture**: We now describe the network architecture that employs the MAA module described above for weakly-supervised temporal action localization. We start from a previous state-of-the-art base architecture, sparse temporal pooling network (STPN) (Nguyen et al., 2017). As shown

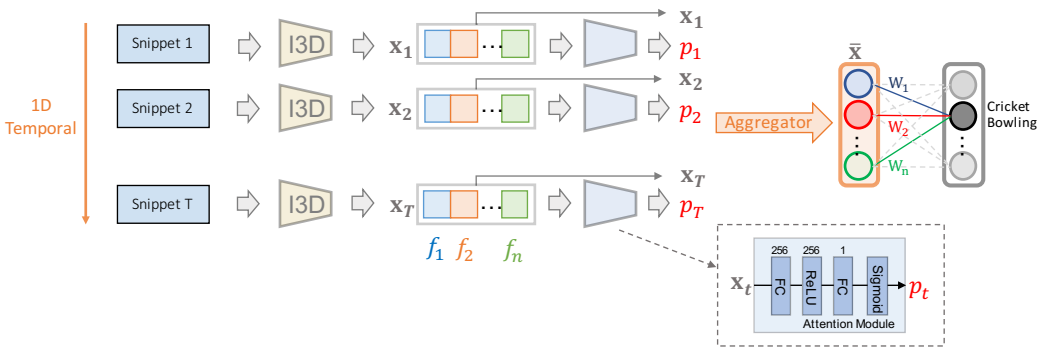


Figure 2: Network architecture for the weakly-supervised action localization.

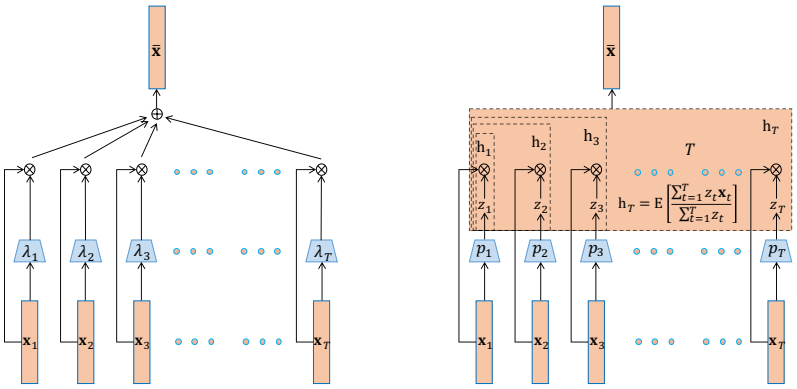

Figure 3: The feature aggregators used in STPN and MAAN.

in Figure. 2, it first divides the input video into several non-overlapped snippets and extracts the I3D (Carreira & Zisserman, 2017) feature for each snippet. Each snippet-level feature is then fed to an attention module to generate attention weight between 0 and 1. STPN then uses a feature aggregator to calculate a weighted sum of the snippet-level features with these class-agnostic attention weights to create a video-level representation, as shown in the left of Figure. 3. The video-level representation is then passed through an FC layer followed by a sigmoid layer to obtain class scores. Our MAAN uses the attention module to generate the latent discriminative probability $p_t$ and replaces the feature aggregator from the weighted sum aggregation to the proposed marginalized average aggregation, which is demonstrated in the right of Figure. 3.

**Training with video-level class labels**: Formally, the architecture first performs aggregation of the snippet-level features (i.e. $\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T$ ) to obtain the video-level representation $\bar{\mathbf{x}}$ ( $\bar{\mathbf{x}} = \mathbb{E}[\sum_{i=1}^{T} z_i \mathbf{x}_i / \sum_{i=1}^{T} z_i]$). Then, it followed by a logistic regression layer (FC layer + sigmoid) to output video-level classification prediction probability. Specifically, the prediction probability for class $c \in \{1, 2, \cdots C\}$ is parameterized as $\sigma_j^c = \sigma(\mathbf{w}_c^\top \bar{\mathbf{x}}_j)$, where $\bar{\mathbf{x}}_j$ is the aggregated feature for video $j \in \{1, ..., N\}$. Suppose each video $\bar{\mathbf{x}}_j$ is i.i.d and each action class is independent from each other, then the negative log-likelihood function (cross-entropy loss) is given as follows:

$$\mathcal{L}(\mathbf{W}) = -\sum_{j=1}^{N} \sum_{c=1}^{C} \left( y_j^c \log \sigma_j^c + (1 - y_j^c) \log(1 - \sigma_j^c) \right) \tag{10}$$

where $y_j^c \in \{0, 1\}$ is the ground-truth video-level label for class $c$ happens in video $j$ and $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_C]$.

**Temporal Action Localization**: Let $s^c = \mathbf{w}_c^\top \bar{\mathbf{x}}$ be the video-level action prediction score, and $\sigma(s^c) = \sigma(\mathbf{w}_c^\top \bar{\mathbf{x}})$ be the video-level action prediction probability. In STPN, as $\bar{\mathbf{x}} = \sum_{t=1}^{T} \lambda_t \mathbf{x}_t$, the

$s^c$ can be rewritten as:

$$s^c = \mathbf{w}_c^\top \overline{\mathbf{x}} = \sum\nolimits_{t=1}^{T} \lambda_t \mathbf{w}_c^\top \mathbf{x}_t, \tag{11}$$

In STPN, the prediction score of snippet $t$ for action class c in a video is defined as:

$$s_t^c = \lambda_t \sigma(\mathbf{w}_c^\top \mathbf{x}_t), \tag{12}$$

where $\sigma(\cdot)$ denotes the sigmoid function. In MAAN, as $\bar{\mathbf{x}} = \mathbb{E}[\sum_{i=1}^{T} z_i \mathbf{x}_i / \sum_{i=1}^{T} z_i]$, according to Proposition 1, the $s^c$ can be rewritten as:

$$s^c = \mathbf{w}_c^\top \overline{\mathbf{x}} = \mathbf{w}_c^\top \mathbb{E}[\sum\nolimits_{i=1}^{T} z_i \mathbf{x}_i / \sum\nolimits_{i=1}^{T} z_i] = \sum\nolimits_{t=1}^{T} c_t p_t \mathbf{w}_c^\top \mathbf{x}_t. \tag{13}$$

The latent discriminative probability $p_t$ corresponds to the class-agnostic attention weight for snippet $t$. According to Proposition 1 and Proposition 2, $c_t$ does not relate to the snippet $t$, but captures the context of other snippets. $\mathbf{w}_c$ corresponds to the class-specific weights for action class $c$ for all the snippets, and $\mathbf{w}_c^\top \mathbf{x}_t$ indicates the relevance of the snippet $t$ to class $c$. To generate temporal proposals, we compute the prediction score of snippet $t$ belonging to action class $c$ in a video as:

$$s_t^c = p_t \sigma(\mathbf{w}_c^\top \mathbf{x}_t). \tag{14}$$

We denote the $\mathbf{s}^c = (s_1^c, s_2^c, ..., s_T^c)\top$ as the class activation sequence (CAS) for class $c$. Similar to STPN, the threshold is applied to the CAS for each class to extract the one-dimensional connected components to generate its temporal proposals, and then we perform non-maximum suppression among temporal proposals of each class independently to remove highly overlapped detections.

Compared to STPN (Eq. (12)), MAAN (Eq. (14)) employs the latent discriminative probability $p_t$ instead of directly using the attention weight $\lambda_t$ (equivalent to $c_t p_t$) for prediction. Proposition 2 suggests that MAAN can suppress the dominant response $s_t^c$ compared with STPN. Thus, MAAN is more promising to achieve better performance in weakly-supervised temporal action localization.

# 3 Experiments

## 3.1 Experimental Settings

**Datasets.** We evaluate MAAN on two popular action localization benchmark datasets, THUMOS14 (Jiang et al., 2014) and ActivityNet1.3 (Heilbron et al., 2015). **THUMOS14** contains 20 action classes for the temporal action localization task. It consists of 200 untrimmed videos (3,027 action instances) in the validation set and 212 untrimmed videos (3,358 action instances) in the test set. Following the standard practice, we train the models on the validation set without using the temporal annotations and evaluate them on the test set. **ActivityNet1.3** is a large-scale video benchmark for action detection that covering a wide range of complex human activities. It provides samples from 200 activity classes with an average of 137 untrimmed videos per class and 1.41 activity instances per video, for a total of 849 video hours. This dataset contains 10,024 training videos, 4,926 validation videos and 5,044 test videos. In the experiments, we train the models on the training videos and test on the validation videos. **Evaluation Metrics.** We follow the standard evaluation metric by reporting mean average precision (mAP) values at several different levels of intersection over union (IoU) thresholds. We use the benchmarking code provided by ActivityNet[1] to evaluate the models.

**Implementation Details.** We use two-stream I3D networks (Carreira & Zisserman, 2017) pre-trained on the Kinetics dataset (Kay et al., 2017) to extract the snippet-level feature vectors for each video. All the videos are divided into a set of non-overlapping video snippets. Each snippet contains 16 consecutive frames or optical flow maps. We input each 16 stacked RGB frames or flow maps into the I3D RGB or flow models to extract the corresponding feature vectors of dimension 1024. Due to the various lengths of the videos, in the training, we uniformly divide each video into $T$ non-overlapped segments, and randomly sample one snippet from each segment. Therefore, we sample $T$ snippets for each video as the input of the model for training. We set $T$ to 20 in our MAAN model. The attention module in Figure. 2 consists of an FC layer of $1024 \times 256$, a LeakyReLU layer, an FC

[1] https://github.com/activitynet/ActivityNet/tree/master/Evaluation

Table 1: Comparison of the proposed MAAN with four baseline feature aggregators on the THUMOS14 test set with two-stream I3D feature. All values are reported in percentage (%). The last column demonstrates the classification mAP.

| Methods | AP@IoU | | | | | | | | | Cls mAP |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | |
| STPN | 57.4 | 48.7 | 40.3 | 29.5 | 19.8 | 11.4 | 5.8 | 1.7 | 0.2 | 94.2 |
| Dropout | 53.4 | 44.9 | 35.4 | 25.0 | 16.2 | 8.7 | 4.3 | 1.3 | 0.1 | 92.4 |
| Norm | 48.0 | 39.9 | 30.5 | 20.9 | 12.3 | 5.7 | 2.4 | 0.6 | 0.1 | **95.2** |
| SoftMaxNorm | 22.2 | 17.2 | 12.8 | 9.6 | 6.3 | 4.3 | 2.8 | 1.0 | 0.1 | 94.8 |
| MAAN | **59.8** | **50.8** | **41.1** | **30.6** | **20.3** | **12.0** | **6.9** | **2.6** | **0.2** | 94.1 |

layer of $256 \times 1$, and a sigmoid non-linear activation, to generate the latent discriminative probability $p_t$. We pass the aggregated video-level representation through an FC layer of $1024 \times C$ followed by a sigmoid activation to obtain class scores. We use the ADAM optimizer (Kingma & Ba, 2014) with an initial learning rate of $5 \times 10^{-4}$ to optimize network parameters. At test time, we first reject classes whose video-level probabilities are below $0.01$. We then forward all the snippets of the video to generate the CAS for the remaining classes. We generate the temporal proposals by cutting the CAS with a threshold $th$. The combination ratio of two-stream modalities is set to $0.5$ and $0.5$. Our algorithm is implemented in PyTorch [2]. We run all the experiments on a single NVIDIA Tesla M40 GPU with 24 GB memory.

### 3.2 THUMOS14 DATASET

We first compare our MAAN model on the THUMOS14 dataset with several baseline models that use different feature aggregators in Figure. 2 to gain some basic understanding of the behavior of our proposed MAA. The descriptions of the four baseline models are listed below.

(1) **STPN.** It employs the weighed sum aggregation $\bar{\mathbf{x}} = \sum_{t=1}^{T} \lambda_t \mathbf{x}_t$ to generate the video-level representation. (2) **Dropout.** It explicitly performs dropout sampling with dropout probability $p = 0.5$ in STPN to obtain the video-level representation, $\bar{\mathbf{x}} = \sum_{t=1}^{T} r_t \lambda_t \mathbf{x}_t$, $r_t \sim Bernoulli(0.5)$. (3) **Normalization.** Denoted as "Norm" in the experiments, it utilizes the weighted average aggregation $\bar{\mathbf{x}} = \sum_{t=1}^{T} \lambda_t \mathbf{x}_t / \sum_{t=1}^{T} \lambda_t$ for the video-level representation. (4) **SoftMax Normalization.** Denoted as "SoftMaxNorm" in the experiments, it applies the softmax function as the normalized weights to get the weighted average aggregated video-level feature, $\bar{\mathbf{x}} = \sum_{t=1}^{T} e^{\lambda_t} \mathbf{x}_t / \sum_{t=1}^{T} e^{\lambda_t}$.

We test all the models with the cutting threshold $th$ as 0.2 of the max value of the CAS. We compare the detection average precision (%) at IoU = [0.1 : 0.1 : 0.9] and the video-level classification mean average precision (%) (denoted as Cls mAP) on the test set in Table 1. From Table 1, we can observe that although all the methods achieve similar video-level classification mAP, their localization performances vary a lot. It shows that achieving a good video-level classification performance cannot guarantee to obtain a good snippet-level localization performance because the former only requires the correct prediction on the existence of an action, while the latter requires the correct prediction on both of its existence and its duration and location. Moreover, Table 1 demonstrates that MAAN consistently outperforms all the baseline models at different levels of IoUs in the weakly-supervised temporal localization task. Both of the "Norm" and "SoftmaxNorm" are the normalized weighted average aggregation. However, the "SoftmaxNorm" performs the worst, because the softmax function over-amplifies the weight of the most salient snippet. As a result, it tends to identify very few discriminative snippets and obtain sparse and non-integral localization. The "Norm" also performs worse than our MAAN. It is the normalized weighted average over the snippet-level representation, while MAAN can be considered as the normalized weighted average (expectation) over the subset-level representation. Therefore, MAAN encourages to identify dense and integral action segments as compared to "Norm" which encourages to identify just several discriminative snippets. MAAN works better than the "Dropout" because the "Dropout" randomly dropouts the snippets with different attention weights by uniform probabilities. At each iteration, the scale of the aggregated feature varies a lot, however, MAAN samples with the learnable latent discriminative probability and conducts

[2] https://github.com/pytorch/pytorch

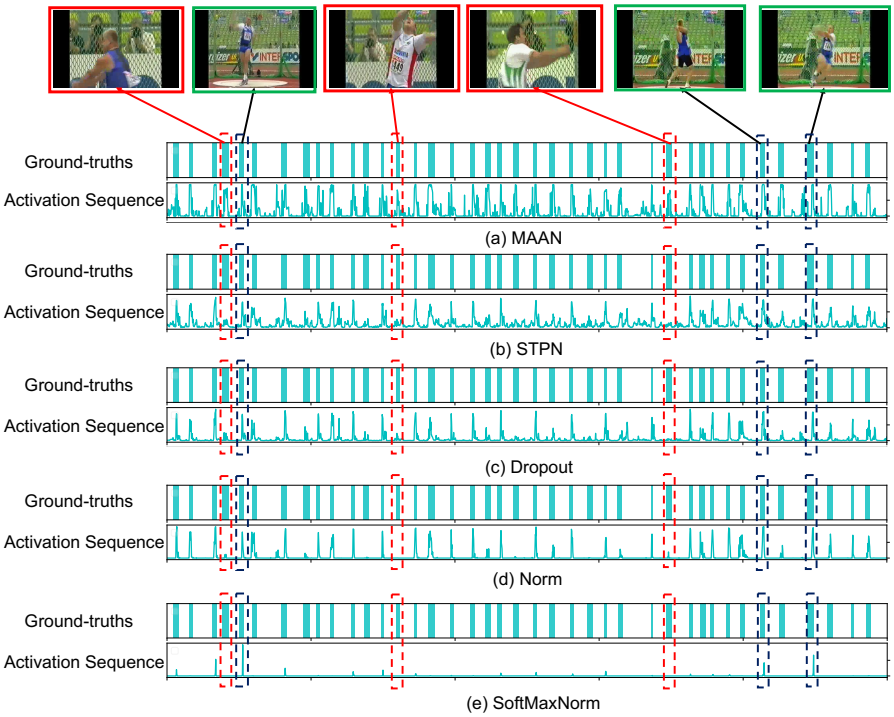


Figure 4: Visualization of the one-dimensional activation sequences on an example of the HammerThrow action in the test set of THUMOS14. The horizontal axis denotes the temporal dimension, which is normalized to [0, 1]. The first row of each model shows the ground-truth action segments. The second row demonstrates the predicted activation sequence for class HammerThrow.

the expectation to keep the scale of the aggregated feature stable. Compared to STPN, MAAN also achieves superior results. MAAN implicitly factorizes the attention weight into $c_t p_t$, where $p_t$ learns the latent discriminative probability for the current snippet, and $c_t$ captures the contextual information and regularizes the network to learn more informative aggregation. The properties of MAA enable the predicted class activation sequences not to concentrate on the most salient regions. The quantitative results demonstrate the effectiveness of the feature aggregation of our proposed MAAN.

Figure. 4 visualize the one-dimensional CASs of the proposed MAAN and all the baseline models. The temporal CAS generated by MAAN can cover large and dense regions to obtain more accurate action segments. In the example in Figure. 4, MAAN can discover almost all the actions compared with the ground-truth; however, the STPN have missed several action segments, and also tends to find the more salient regions in each action segment. Other methods are much sparser compared with MAAN. The first row of the Figure. 4 shows several action segments in red and in green, corresponding to action segments that are relatively difficult and easy to be localized, respectively. We can see that all the easily-localized segments contain the whole person that performing the "HammerThrow" action, while the difficultly-localized segments contain only a part of the person or the action. Our MAAN can successfully localize the easy segments as well as the difficult segments; however, all the other methods fail on the difficult ones. It shows that MAAN can identify several dense and integral action regions other than only the most discriminative region identified by the other methods in comparison.

We also compare our model with the state-of-the-art action localization approaches on THUMOS14 dataset. The numerical results are summarized in Table. 2. We include both fully and weakly-supervised learning, as in (Nguyen et al., 2017). As shown in Table. 2, our implemented STPN performs slightly better than the results reported in the original paper (Nguyen et al., 2017). From Table. 2, our proposed MAAN outperforms the STPN and most of the existing weakly-supervised action localization approaches. Furthermore, even trained with only video-level labels, our model still presents competitive results compared with several recent fully-supervised approaches.

Table 2: Comparison of our algorithm with the previous approaches on THUMOS14 test set. AP (%) is reported for different IoU threshold. Both the fully supervised and the weakly-supervised results are listed. ("UN" means using UntrimmedNet features, "I3D" means using I3D features, "ours" means our implementation.)

| Supervision | Methods | AP@IoU | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Fully Supervised | Richard et al. (Richard & Gall, 2016) | 39.7 | 35.7 | 30.0 | 23.2 | 15.2 | - | - | - | - |
| | Shou et al. (Shou et al., 2016) | 47.7 | 43.5 | 36.3 | 28.7 | 19.0 | 10.3 | 5.3 | - | - |
| | Yeung et al. (Yeung et al., 2016) | 48.9 | 44.0 | 36.0 | 26.4 | 17.1 | - | - | - | - |
| | Yuan et al. (Yuan et al., 2016) | 51.4 | 42.6 | 33.6 | 26.1 | 18.8 | - | - | - | - |
| | Shou et al. (Shou et al., 2017) | - | - | 40.1 | 29.4 | 23.3 | 13.1 | 7.9 | - | - |
| | Yuan et al. (Yuan et al., 2017b) | 51.0 | 45.2 | 36.5 | 27.8 | 17.8 | - | - | - | - |
| | Xu et al. (Xu et al., 2017) | 54.5 | 51.5 | 44.8 | 35.6 | 28.9 | - | - | - | - |
| | Zhao et al. (Zhao et al., 2017) | 66.0 | 59.4 | 51.9 | 41.0 | 29.8 | - | - | - | - |
| Weakly Supervised | Wang et al. (Wang et al., 2017) | 44.4 | 37.7 | 28.2 | 21.1 | 13.7 | - | - | - | - |
| | Singh & Lee (Singh & Lee, 2017) | 36.4 | 27.8 | 19.5 | 12.7 | 6.8 | - | - | - | - |
| | STPN (Nguyen et al., 2017) (UN) | 45.3 | 38.8 | 31.1 | 23.5 | 16.2 | 9.8 | 5.1 | 2.0 | **0.3** |
| | STPN (Nguyen et al., 2017) (I3D) | 52.0 | 44.7 | 35.5 | 25.8 | 16.9 | 9.9 | 4.3 | 1.2 | 0.1 |
| | STPN (Nguyen et al., 2017) (ours) | 57.4 | 48.7 | 40.3 | 29.5 | 19.8 | 11.4 | 5.8 | 1.7 | 0.2 |
| | AutoLoc (Shou et al., 2018) | - | - | 35.8 | 29.0 | **21.2** | **13.4** | 5.8 | - | - |
| | **MAAN (ours)** | **59.8** | **50.8** | **41.1** | **30.6** | 20.3 | 12.0 | **6.9** | **2.6** | 0.2 |

Table 3: Comparison of our algorithm with the state-of-the-art approaches on ActivityNet1.3 validation set. AP (%) is reported for different IoU threshold $\alpha$. ("ours" means our implementation.)

| Supervision | Methods | AP @ IoU | | |
|---|---|---|---|---|
| | | 0.5 | 0.75 | 0.95 |
| Fully-supervised | Singh & Cuzzolin (Singh & Cuzzolin, 2016) | 34.5 | - | - |
| | Wang & Tao (Wang & Tao, 2016) | 45.1 | 4.1 | 0.0 |
| | Shou et al. (Shou et al., 2017) | 45.3 | 26.0 | 0.2 |
| | Xiong et al. (Xiong et al., 2017) | 39.1 | 23.5 | 5.5 |
| Weakly-supervised | STPN (Nguyen et al., 2017) | 29.3 | 16.9 | 2.6 |
| | STPN (Nguyen et al., 2017) (ours) | 29.8 | 17.7 | 4.1 |
| | **MAAN (ours)** | **33.7** | **21.9** | **5.5** |

### 3.3 ACTIVITYNET1.3 DATASET

We train the MAAN model on the ActivityNet1.3 training set and compare our performance with the recent state-of-the-art approaches on the validation set in Table. 3. The action segment in ActivityNet is usually much longer than that of THUMOS14 and occupies a larger percent of a video. We use a set of thresholds, which are $[0.2, 0.15, 0.1, 0.05]$ of the max value of the CAS, to generate the proposals from the one-dimensional CAS. As shown in Table. 3, with the set of thresholds, our implemented STPN performs slightly better than the results reported in the original paper (Nguyen et al., 2017). With the same threshold and experimental setting, our proposed MAAN model outperforms the STPN approach on the large-scale ActivityNet1.3. Similar to THUMOS14, our model also achieves good results that are close to some of the fully-supervised approaches.

## 4 CONCLUSION

We have proposed the marginalized average attentional network (MAAN) for weakly-supervised temporal action localization. MAAN employs a novel marginalized average aggregation (MAA) operation to encourage the network to identify the dense and integral action segments in an end-to-end fashion. Theoretically, we have proved that MAA decreases the gap between the most discriminant regions in the video to the others, and thus MAAN generates better class activation sequences to infer the action locations. We have also proposed a fast algorithm to reduce the computation complexity of MAA. Our proposed MAAN achieves superior performance on both of the THUMOS14 and the ActivityNet1.3 dataset on weakly-supervised temporal action localization task.

## REFERENCES


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48. ACM, 2009.

Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016.

J. Carreira and A Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):189–203, 2017.

Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems*, pp. 33–44, 2017.

Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r* cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1080–1088, 2015.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1025–1035, 2017.

F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pp. 1693–1701, 2015.

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://crcv.ucf.edu/THUMOS14/`, 2014.

Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. ContextLocNet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2016.

W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Greem, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. In *arXiv:1705.06950v1*, 2017.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. Structured attention networks. *arXiv preprint arXiv:1702.00887*, 2017.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*, 2014.

Shu Kong and Charless Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7025–7034. IEEE, 2017.

Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. *arXiv preprint arXiv:1802.03676*, 2018.

Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. *CVPR*, 2017.

Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 685–694, 2015.

Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1713–1721, 2015.

Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3131–3140, 2016.

Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015.

Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1049–1058, 2016.

Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017.

Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 154–171, 2018.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pp. 568–576, 2014.

Gurkirt Singh and Fabio Cuzzolin. Untrimmed video classification for activity detection: submission to activitynet challenge. *arXiv preprint arXiv:1607.01979*, 2016.

Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pp. 20–36. Springer, 2016.

Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. *CVPR*, 2017.

R. Wang and D. Tao. Acitivitynet large scale activity recognition challenge. *UTS at Activitynet*, 2016.

Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE CVPR*, 2017.

Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017.

H. A. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017.

Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2678–2687, 2016.

J. Yuan, B. Ni, X. Yang, and A. A. Kassim. Temporal action localization with pyramid of score distribution features. In *CVPR*, 2016.

Yuan Yuan, Xiaodan Liang, Xiaolong Wang, Dit-Yan Yeung, and Abhinav Gupta. Temporal dynamic graph LSTM for action-driven video object detection. In *ICCV*, pp. 1819–1828, 2017a.

Z. Yuan, J. Stroud, T. Lu, and J. Deng. Temporal action localization by structured maximal sums. In *CVPR*, 2017b.

Dingwen Zhang, Deyu Meng, and Junwei Han. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE transactions on pattern analysis and machine intelligence*, 39 (5):865–878, 2017.

Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*, 2018a.

Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas Huang. Adversarial complementary learning for weakly supervised object localization. *arXiv preprint arXiv:1804.06962*, 2018b.

Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.

B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. *arXiv preprint arXiv:1709.01829*, 2017.

# A PROOF OF PROPOSITION 1

## A.1 PROOF OF EQUATION (3)

*Proof.*

$$\mathbb{E}\left[\frac{\sum_{i=1}^{T} z_i \mathbf{x}_i}{\sum_{i=1}^{T} z_i}\right] = \sum\nolimits_{i=1}^{T} \mathbb{E}[z_i / \sum\nolimits_{i=1}^{T} z_i]\mathbf{x}_i \tag{15}$$

In addition,

$$\mathbb{E}[z_i / \sum\nolimits_{i=1}^{T} z_i] = p_i \times \mathbb{E}\left[1/(1+\sum\nolimits_{k=1,k\neq i}^{T} z_k)\right] + (1-p_i)\times 0 = p_i c_i \tag{16}$$

Thus , we achieve that

$$\mathbb{E}\left[\frac{\sum_{i=1}^{T} z_i \mathbf{x}_i}{\sum_{i=1}^{T} z_i}\right] = \sum\nolimits_{i=1}^{T} c_i p_i \mathbf{x}_i = \sum\nolimits_{i=1}^{T} \lambda_i \mathbf{x}_i \tag{17}$$

□

## A.2 PROOF OF $p_i \geq p_j \Leftrightarrow c_i \geq c_j \Leftrightarrow \lambda_i \geq \lambda_j$

*Proof.* Denote $S_T = \sum_{k=1,k\neq i,k\neq j}^{T} z_k$, then we have that

$$c_i - c_j = \mathbb{E}\left[1/(1+\sum\nolimits_{k\neq i} z_k)\right] - \mathbb{E}\left[1/(1+\sum\nolimits_{k\neq j} z_k)\right] \tag{18}$$

$$= p_j \mathbb{E}\left[1/(2+S_T)\right] + (1-p_j)\mathbb{E}\left[1/(1+S_T)\right] - p_i \mathbb{E}\left[1/(2+S_T)\right] - (1-p_i)\mathbb{E}\left[1/(1+S_T)\right]$$

$$= (p_i - p_j)\left(\mathbb{E}\left[1/(1+S_T)\right] - \mathbb{E}\left[1/(2+S_T)\right]\right) \tag{19}$$

Since $\mathbb{E}\left[1/(1+S_T)\right] - \mathbb{E}\left[1/(2+S_T)\right] > 0$, we achieve that $p_i \geq p_j \Leftrightarrow c_i \geq c_j$. Since $\lambda_i = c_i p_i$ and $\lambda_j = c_j p_j$, and $c_i, c_j, p_i, p_j \geq 0$, it follows that $p_i \geq p_j \Leftrightarrow \lambda_i \geq \lambda_j$

□

# B PROOF OF PROPOSITION 2

*Proof.* $\sum_{i=1}^{T} c_i p_i = \sum_{i=1}^{T} \mathbb{E}[z_i / \sum_{i=1}^{T} z_i] = \mathbb{E}\left[(\sum_{i=1}^{T} z_i)/(\sum_{i=1}^{T} z_i)\right] = 1$

When $p_1 = p_2 = \cdots = p_T$, we have $\lambda_1 = \lambda_2 = \cdots = \lambda_T$. then inequality (4) trivially holds true. Without loss of generality, assume $p_1 \geq p_2 \geq \cdots \geq p_T$ and there exists a strict inequality. Then $\exists k \in \{1, ..., T-1\}$ such that $c_i \geq 1/(\sum_{t=1}^{T} p_t)$ for $1 \leq i \leq k$ and $c_j \leq 1/(\sum_{t=1}^{T} p_t)$ for $k < j \leq T$. Otherwise, we obtain $c_i \geq 1/(\sum_{t=1}^{T} p_t)$ or $c_i \leq 1/(\sum_{t=1}^{T} p_t)$ for $1 \leq i \leq T$ and there exists a strict inequality. It follows that $\sum_{i=1}^{T} c_i p_i > 1$ or $\sum_{i=1}^{T} c_i p_i < 1$, which contradicts with $\sum_{i=1}^{T} c_i p_i = 1$. Thus, we obtain the set $\mathcal{I} \neq \emptyset$.

Without loss of generality, for $1 \leq i \leq k$ and $i \leq j \leq T$ , we have $c_i \geq 1/(\sum_{t=1}^{T} p_t)$ and $p_i \geq p_j$, then we obtain that $c_i \geq c_j$. It follows that

$$p_i/(\sum\nolimits_{t=1}^{T} p_t) - p_j/(\sum\nolimits_{t=1}^{T} p_t) - \left(\lambda_i/(\sum\nolimits_{t=1}^{T} \lambda_t) - \lambda_j/(\sum\nolimits_{t=1}^{T} \lambda_t)\right) \tag{20}$$

$$= p_i/(\sum\nolimits_{t=1}^{T} p_t) - p_j/(\sum\nolimits_{t=1}^{T} p_t) - (c_i p_i - c_j p_j) \tag{21}$$

$$= \left(1/(\sum\nolimits_{t=1}^{T} p_t) - c_i\right) p_i - \left(1/(\sum\nolimits_{t=1}^{T} p_t) - c_j\right) p_j \tag{22}$$

$$\leq \left(1/(\sum\nolimits_{t=1}^{T} p_t) - c_i\right) p_i - \left(1/(\sum\nolimits_{t=1}^{T} p_t) - c_i\right) p_j \tag{23}$$

$$= \left(1/(\sum\nolimits_{t=1}^{T} p_t) - c_i\right)(p_i - p_j) \leq 0 \tag{24}$$

□

# C PROOF OF PROPOSITION 3

## C.1 COMPUTATION OF $\mathbf{h}_t$

$$\mathbf{h}_t = E[\frac{\mathbf{Y_t}}{Z_t}] = \sum_{z_1,z_2,\ldots,z_t} P\left(z_1, z_2, \cdots z_t\right) \frac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{\sum_{j=1}^{t} z_j} \tag{25}$$

$$= \sum\nolimits_{i=0}^{t} \left( \sum_{z_1,z_2,\cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i\right) P\left(z_1, z_2, ..., z_t\right) \frac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{\sum_{j=1}^{t} z_j} \right) \tag{26}$$

$$= \sum\nolimits_{i=0}^{t} \sum_{z_1,z_2,\ldots,z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i\right) P\left(z_1, z_2, \cdots z_t\right) \frac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{i} \tag{27}$$

$$= \sum\nolimits_{i=0}^{t} \mathbf{m}_i^t \tag{28}$$

where $\mathbf{1}(\cdot)$ denotes the indicator function.

We achieve Eq.(26) by partitioning the summation into $t+1$ groups . Terms belong to group $i$ have $\sum_{j=1}^{t} z_j = i$.

Let $\mathbf{m}_i^t = \sum_{z_1,z_2,\cdots z_t} \mathbf{1}\left(\sum_{j=1}^{t} z_j = i\right) P\left(z_1, z_2, \cdots z_t\right) \frac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{i}$, and we achieve Eq.(28)

## C.2 PROOF OF RECURRENT FORMULA OF $m_i^{t+1}$

We now give the proof of recurrent formula of Eq.(29)

$$\mathbf{m}_i^{t+1} = p_{t+1}\left(b_{i-1}\mathbf{m}_{i-1}^t + (1-b_{i-1})q_{i-1}^t \mathbf{x}_{t+1}\right) + (1-p_{t+1})\mathbf{m}_i^t \tag{29}$$

*Proof.*

$$\mathbf{m}_i^{t+1} = \sum_{z_1,z_2,\cdots z_t,z_{t+1}} \mathbf{1}\left(\sum\nolimits_{j=1}^{t+1} z_j = i\right) P\left(z_1, z_2, \cdots z_{t+1}\right) \frac{\sum_{j=1}^{t+1} z_j \mathbf{x}_j}{i} \tag{30}$$

$$= \sum_{z_1,z_2,\cdots z_t,z_{t+1}} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j + z_{t+1} = i\right) P\left(z_1, z_2, \cdots z_t\right) P(z_{t+1}) \frac{\sum_{j=1}^{t} z_j \mathbf{x}_j + z_{t+1}\mathbf{x}_{t+1}}{i} \tag{31}$$

$$\begin{aligned} = & \sum_{z_1,z_2,\cdots z_t} \left[\mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j + 1 = i\right) P\left(z_1, z_2, \cdots z_t\right) p_{t+1} \tfrac{\sum_{j=1}^{t} z_j \mathbf{x}_j + \mathbf{x}_{t+1}}{i}\right] \\ & + \sum_{z_1,z_2,\cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i\right) P\left(z_1, z_2, \cdots z_t\right) (1-p_{t+1}) \tfrac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{i} \end{aligned} \tag{32}$$

$$\begin{aligned} = & \sum_{z_1,z_2,\cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j + 1 = i\right) P\left(z_1, z_2, \cdots z_t\right) p_{t+1} \tfrac{\sum_{j=1}^{t} z_j \mathbf{x}_j + \mathbf{x}_{t+1}}{i} \\ & + (1-p_{t+1}) \sum_{z_1,z_2,\cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i\right) P\left(z_1, z_2, \cdots z_t\right) \tfrac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{i} \end{aligned} \tag{33}$$

$$\begin{aligned} = & p_{t+1} \sum_{z_1,z_2,\cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i-1\right) P\left(z_1, z_2, \cdots z_t\right) \tfrac{i-1}{i} \tfrac{\sum_{j=1}^{t} z_j \mathbf{x}_j + \mathbf{x}_{t+1}}{i-1} \\ & + (1-p_{t+1}) m_i^t \end{aligned} \tag{34}$$

$$\begin{aligned} = & p_{t+1} \sum_{z_1,z_2,\cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i-1\right) P\left(z_1, z_2, \cdots z_t\right) \left[\tfrac{i-1}{i} \tfrac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{i-1} + \tfrac{\mathbf{x}_{t+1}}{i}\right] \\ & + (1-p_{t+1}) \mathbf{m}_i^t \end{aligned} \tag{35}$$

$$\begin{aligned} = & p_{t+1} \sum_{z_1,z_2,\cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i-1\right) P\left(z_1, z_2, \cdots z_t\right) \left[b_{i-1} \tfrac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{i-1} + (1-b_{i-1})\mathbf{x}_{t+1}\right] \\ & + (1-p_{t+1}) \mathbf{m}_i^t \end{aligned} \tag{36}$$

Then, we have

$$\mathbf{m}_i^{t+1} = \begin{array}{l} p_{t+1} b_{i-1} \sum\limits_{z_1, z_2, \cdots z_t} \mathbf{1}\left(\sum_{j=1}^{t} z_j = i-1\right) P\left(z_1, z_2, \cdots z_t\right) \frac{\sum_{j=1}^{t} z_j \mathbf{x}_j}{i-1} \\ + p_{t+1}(1-b_{i-1}) \sum\limits_{z_1, z_2, \cdots z_t} \mathbf{1}\left(\sum_{j=1}^{t} z_j = i-1\right) P\left(z_1, z_2, \cdots z_t\right) \mathbf{x}_{t+1} + (1-p_{t+1}) \mathbf{m}_i^t \end{array} \tag{37}$$

Since $q_{i-1}^t = P\left(\sum_{j=1}^{t} z_j = i-1\right) = \sum\limits_{z_1, z_2, \cdots z_t} \mathbf{1}\left(\sum_{j=1}^{t} z_j = i-1\right) P\left(z_1, z_2, \cdots z_t\right)$ we can achieve that

$$\mathbf{m}_i^{t+1} = p_{t+1}\left[b_{i-1}\mathbf{m}_{i-1}^t + (1-b_{i-1}) q_{i-1}^t \mathbf{x}_{t+1}\right] + (1-p_{t+1})\mathbf{m}_i^t \tag{38}$$

□

### C.3 Proof of recurrent formula of $q_i^{t+1}$

We present the proof of Eq.(39)

$$q_i^{t+1} = p_{t+1} q_{i-1}^t + (1-p_{t+1}) q_i^t \tag{39}$$

*Proof.*

$$\begin{aligned} q_i^{t+1} &= \sum_{z_1, z_2, \cdots z_t, z_{t+1}} \mathbf{1}\left(\sum\nolimits_{j=1}^{t+1} z_j = i\right) P\left(z_1, z_2, \cdots z_{t+1}\right) && (40) \\ &= \sum_{z_1, z_2, \cdots z_t, z_{t+1}} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j + z_{t+1} = i\right) P\left(z_1, z_2, \cdots z_t\right) P(z_{t+1}) && (41) \\ &= \sum_{z_1, z_2, \cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j + 1 = i\right) P\left(z_1, z_2, \cdots z_t\right) p_{t+1} && (42) \\ &+ \sum_{z_1, z_2, \cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i\right) P\left(z_1, z_2, \cdots z_t\right) (1-p_{t+1}) && (43) \\ &= p_{t+1} \sum_{z_1, z_2, \cdots z_t} \mathbf{1}\left(\sum\nolimits_{j=1}^{t} z_j = i-1\right) P\left(z_1, z_2, \cdots z_t\right) + (1-p_{t+1}) q_i^t && (44) \\ &= p_{t+1} q_{i-1}^t + (1-p_{t+1}) q_i^t && (45) \end{aligned}$$

□

## D Related Work

**Video Action Analysis.** Researchers have developed quite a few deep networks for video action analysis. Two-stream networks (Simonyan & Zisserman, 2014) and 3D convolutional neural networks (C3D) (Tran et al., 2015) are popular solutions to learn video representations and these techniques, including their variations, are extensively used for video action analysis. Recently, a combination of two-stream networks and 3D convolutions, referred to as I3D (Carreira & Zisserman, 2017), was proposed as a generic video representation learning method, and served as an effective backbone network in various video analysis tasks such as recognition (Wang et al., 2016), localization (Shou et al., 2016), and weakly-supervised learning (Wang et al., 2017).

**Weakly-Supervised Temporal Action Localization.** There are only a few approaches based on weakly-supervised learning that rely solely on video-level class labels to localize actions in the temporal domain. Wang et al. (Wang et al., 2017) proposed a UntrimmedNet framework, where two softmax functions are applied across class labels and proposals to perform action classification and detect important temporal segments, respectively. However, using the softmax function across proposals may not be effective to identify multiple instances. Singh et al. (Singh & Lee, 2017)

designed a Hide-and-Seek model to randomly hide some regions in a video during training and force the network to seek other relevant regions. However, the randomly hiding operation, as a data augmentation, cannot guarantee whether it is the action region or the background region that be hidden during training, especially when the dropout probabilities for all the regions are the same. Nguyen et al. (Nguyen et al., 2017) proposed a sparse temporal pooling network (STPN) to identify a sparse set of key segments associated with the actions through attention-based temporal pooling of video segments. However, the sparse constraint may force the network to focus on very few segments and lead to incomplete detection segments. In order to avoid concentrating only on the most salient regions, we are inspired to propose the MAAN model to explicitly take the expectation with respect to average aggregated features of all the sampled subsets from the video in an end-to-end fashion.

**Feature Aggregators.** Learning discriminative localization representations with only video-level class labels requires the feature aggregation operation to turn the multiple snippet-level representations into a video-level representation for classification. Feature aggregation mechanism is widely adopted in deep learning literature and various of tasks and scenarios, for example, neural machine translation (Bahdanau et al., 2015), visual question answering (Hermann et al., 2015), and so on. However, most of these cases are fully-supervised learning where the goal is to learn a model that attends the most relevant features with the supervision information corresponding to the task directly. Many variant feature aggregators have been proposed, ranging from the non-parametric max pooling and average pooling, to parametric hard attention (Gkioxari et al., 2015), soft attention (Vaswani et al., 2017; Sharma et al., 2015), second-order pooling (Girdhar & Ramanan, 2017; Kong & Fowlkes, 2017), structured attention (Kim et al., 2017; Mensch & Blondel, 2018), graph aggregators (Zhang et al., 2018a; Hamilton et al., 2017), and so on. Different from the fully-supervised setting, where the feature aggregator is designed for its specific task, we develop a feature aggregator that is trained with class labels, and then it is used to predict the dense action locations for test data. Different from the heuristic approaches (Wei et al., 2017; Zhang et al., 2018b), which can be considered as a kind of hard-code attention by erasing some regions with a hand-crafted threshold, we introduce the end-to-end differentiable marginalized average aggregation which incorporates learnable latent discriminative probabilities into the learning process.

## E MARGINALIZED AVERAGE AGGREGATION

**Algorithm 1** Marginalized Average Aggregation

**Input:** Feature Representations $\{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T\}$ , Sampling Probability $\{p_1, p_2, \cdots p_T\}$.
**Output:** Aggregated Representation $\overline{\mathbf{x}}$
Initialize $\mathbf{m}_0^0 = \mathbf{0}$, $q_0^0 = 1$, $b_i = \frac{i}{i+1}$;
**for** $t = 1$ **to** $T$ **do**
  Set $\mathbf{m}_0^t = \mathbf{0}$, and $q_{-1}^t = 0$ and $q_{t+1}^t = 0$;
  **for** $i = 1$ **to** $t$ **do**
    $q_i^t = p_t q_{i-1}^{t-1} + (1 - p_t)\, q_i^{t-1}$
    $\mathbf{m}_i^t = p_t \left(b_{i-1}\mathbf{m}_{i-1}^{t-1} + (1 - b_{i-1}) q_{i-1}^{t-1}\mathbf{x}_t\right) + (1 - p_t)\mathbf{m}_i^{t-1}$
  **end for**
**end for**
Return $\overline{\mathbf{x}} = \sum_{i=0}^{T} \mathbf{m}_i^T$