Weakly Supervised Action Selection Learning in Video

Junwei Ma* Layer6 AI jeremy@layer6.ai

Satya Krishna Gorti* Layer6 AI satya@layer6.ai

Maksims Volkovs Layer6 AI maks@layer6.ai

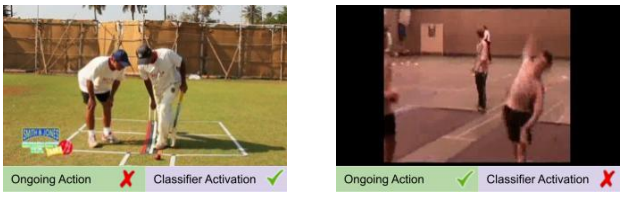
Guangwei Yu Layer6 AI guang@layer6.ai

Abstract

Localizing actions in video is a core task in computer vision. The weakly supervised temporal localization problem investigates whether this task can be adequately solved with only video-level labels, significantly reducing the amount of expensive and error-prone annotation that is required. A common approach is to train a frame-level classifier where frames with the highest class probability are selected to make a video-level prediction. Frame-level activations are then used for localization. However, the absence of frame-level annotations cause the classifier to impart class bias on every frame. To address this, we propose the Action Selection Learning (ASL) approach to capture the general concept of action, a property we refer to as "actionness". Under ASL, the model is trained with a novel class-agnostic task to predict which frames will be selected by the classifier. Empirically, we show that ASL outperforms leading baselines on two popular benchmarks THUMOS-14 and ActivityNet-1.2, with 10.3% and 5.7% relative improvement respectively. We further analyze the properties of ASL and demonstrate the importance of actionness. Full code for this work is available here: https://github.com/layer6ai-labs/ASL.

1. Introduction

Temporal action localization is a fundamental task in computer vision with important applications in video understanding and modelling. The weakly supervised localization problem investigates whether this task can be adequately solved with only video-level labels instead of frame-level annotations. This significantly reduces the expensive and error-prone labelling required in the fully supervised setting [39, 29], but considerably increases the difficulty of the problem. A standard approach is to apply multiple instance learning to learn a classifier over instances, where an instance is typically a frame or a short segment [23, 26, 14]. The classifier is trained using the top-k aggregation over the instance class activation sequence to generate video probabilities. Localization is then done by leveraging the class activation sequence to generate start and end predictions. However, in many cases, the instances that are selected in the top-k contain useful information for prediction but not the actual action. Furthermore, with top-k selection the classification loss encourages the classifier to ignore action instances that are difficult to classify. Both of these problems can significantly hamper localization accuracy and stem from the general inability of the classifier to capture the intrinsic property of action in instances. This property is known as "actionness" in the existing literature [7, 21].



(a) Context error (b) Actionness error

Figure 1: (a) Context error for the "Cricket Shot" action due to the presence of all cricket artifacts but absence of action. (b) Actionness error for the "Cricket Bowling" action due to the atypical scene despite the presence of action.

Ignoring actionness can lead to two major types of error: *context error* and *actionness error*. Context error occurs when the classifier activates on instances that do not contain actions but contain context indicative of the overall video class [19, 14]. Figure 1 (a) shows an example of context error. Here, cricket players are inspecting a cricket pitch. The instance clearly indicates that the video is about cricket and the classifier predicts "Cricket Shot" with high confidence. However, no shot happens in this particular instance and including it in the localization for "Cricket Shot" would lead to an error. Actionness error occurs when the classifier fails to activate on instances that contain actions. This generally occurs in difficult instances that have significant occlusion or uncommon settings. An example of this is shown in Figure 1 (b). The action is "Cricket Bowling", but the classifier

*Authors contributed equally to this work.

fails to activate as the scene is indoors and differs from the usual cricket setting.

Leading recent work [25, 14, 24] in this area propose an attention model to filter out background and then train a classifier on the filtered instances to predict class probabilities. This has the drawback of making it more challenging for the classifier to learn as important context is potentially removed as background.

Our motivation is to design a learning framework that can use the context information for class prediction and at the same time learn to identify action instances for localization. We have seen from the supervised setting that leading object detection [9, 27, 4] and temporal localization [17, 18, 16] methods leverage class-agnostic proposal networks to generate highly accurate predictions. This demonstrates that a general objectness/actionness property can be successfully learned across a diverse set of classes. To this end, we propose a new approach called **A**ction **S**election **L**earning (ASL) where the class-agnostic actionness model learns to predict which frames will be selected in the top-k sets by the classifier. During inference, we combine predictions from the actionness model with class activation sequence and show considerable improvement in localization accuracy. Specifically, ASL achieves new state-of-the-art on two popular benchmarks THUMOS-14 and ActivityNet-1.2, where we improve over leading baselines by 10.3% and 5.7% in mAP respectively. We further analyze the performance of our model and demonstrate the advantages of the actionness approach.

2. Related Work

Weakly Supervised Temporal Action Localization A prominent direction in the weakly supervised setting is to leverage the class activation sequence to improve localization. UntrimmedNet [32] focuses on improving the instance selection step using class activations. Hide-and-seek [30] applies instance dropout to reduce classifier's dependence on specific instances. W-TALC [26] incorporates a co-activity similarity loss to capture inter-class and inter-video relationships. 3C-Net [23] adopts a center loss to reduce inter-class variations while applying additional action-count information for supervision. Focusing on class activations can be susceptible to context error, and a parallel line of research explores how to identify context instances. STPN [24] extends UntrimmedNet by introducing a class-agnostic attention model with sparsity constraints. BM [24] uses self-attention to separate action and context instances. CMCS [19] assumes a stationary prior on context and leverages it to model context instances. BaSNet [14] explicitly models a separate context class that is used to filter instances during inference. DGAM [28] trains a variational autoencoder to model the class-agnostic instance distribution conditioned on attention to separate context from actions. More recently, TSCN [37] and EM-MIL [22] propose two-stream architectures. TSCN separates the RGB and Flow modules and learns from pseudo labels generated by combining the predictions of the two streams. EM-MIL introduces a key instance and a classification module trained alternately to maintain the multi instance learning assumption.

Actionness Learning Our approach is motivated by related work in the supervised setting where a common design choice is to learn a class-agnostic module to generate proposals that are then labelled by the classifier [17, 18, 16]. Earlier work defines actionness as a likelihood of a generic but deliberate action that is separate from context [7], and applies it to detect human activity in both image [7] and video [34] settings. A related concept of "interestingness" has been proposed to identify actions at the pixel level [31]. Work in action recognition shows that generic attributes exist across action classes and can be leveraged for recognition [21]. Similar concept has been demonstrated to be successful for tracking applications [15]. Finally, in object detection, leading approaches heavily leverage class-agnostic proposal networks to first identify regions of high "objectness" [9, 27, 4]. In this work, we demonstrate that the analogous "actionness" property in videos can be effectively learned with only video-level labels.

3. Approach

We treat a video as a set of T instances $\{x_1, ..., x_T\}$, dropping video index to reduce notation clutter. An instance can be a frame or a fixed-interval segment, represented by a feature vector $x_t \in \mathbb{R}^d$. In the weakly supervised temporal localization task, each instance x_t either contains an action from one of C classes or is the background, however, this is unknown to us. Instead, we are given video-level classes $Y \subseteq \{1, ..., C\}$ which is the union of all instance classes in the video. The weakly supervised temporal localization task then asks whether video-level class information can be used to localize actions across instances. In this section, we first outline the classification framework in Section 3.1, and then describe our approach in Section 3.2.

3.1. Base Classifier

We define a video classifier to predict target video-level classes as:

$$s_{c,t} = F_{c,t}(x_1, ..., x_T) \tag{1}$$

where F is a neural network applied to the entire video, and $F_{c,t}(\cdot)$ denotes its output at class c and instance x_t. Taken over all T instances we refer to $F_{c,t}(\cdot)$ as the class activation sequence (CAS). Multiple instance learning [5] is commonly used to train the classifier, where top-k pooling is applied over CAS for each class to aggregate the highest

activated instances and make video-level predictions. We denote the set of top-k instances for each class as $\mathcal{T}^c$:

$$\mathcal{T}^c = \underset{\substack{\mathcal{T}\subseteq\{1,...,T\}\\|\mathcal{T}|=k}}{\arg\max} \sum_{t\in\mathcal{T}} h_{c,t} \tag{2}$$

where k is a hyper-parameter and $h_{c,t}$ is the instance selection probability that is used to select the top instances. In prior work, the selection probability is straightforwardly set to the CAS with $h_{c,t} = s_{c,t}$. However, we make a deliberate distinction here which allows incorporating actionness as we demonstrate in the following section. Aggregation, such as mean pooling, is applied over the selected instances in $\mathcal{T}^c$ to make video-level class prediction:

$$p_c = \text{softmax}\left(\frac{1}{|\mathcal{T}^c|}\sum_{t\in\mathcal{T}^c} s_{c,t}\right) \tag{3}$$

Finally, this model is optimized with the multiple instance learning objective:

$$\mathcal{L}_{\text{CLS}} = -\frac{1}{|Y|}\sum_{c\in Y}\log p_c \tag{4}$$

3.2. Action Selection Learning in Video

The classifier introduced in the previous section optimizes the classification objective which encourages only instances that strongly support the target video classes to get selected in the top-k set. This can lead to the inclusion of instances that provide strong context support but do not contain the action (actionness error), and also the exclusion of instances that contain the action but are difficult to predict (context error). Both of these problems do not affect video classification accuracy, but can significantly hurt localization. We address this by developing a novel action selection learning (ASL) approach to capture the class-agnostic actionness property of each instance. The main idea behind ASL is that the top-k set $\mathcal{T}^c$ used for prediction is likely capturing both context and action instances. However, context information is highly class-specific whereas actions share similar characteristics across classes. Consequently, by training a separate class-agnostic model to predict whether an instance will be in the top-k set for any class, we can effectively capture instances that contain actions and filter out context. We begin by defining a neural network actionness model G:

$$a_t = \sigma\left(G_t(x_1, ..., x_T)\right) \tag{5}$$

where σ is the sigmoid activation function, and $G_t(\cdot)$ denotes the output of G for instance x_t. Here, a_t can be interpreted as the probability that x_t contains any action.

For an instance to contain a specific action, it should simultaneously contain evidence of the corresponding class and evidence of actionness. As we discussed, class evidence alone is not sufficient and can lead to context and actionness errors. To account for both properties, we expand the instance selection function:

$$h_{c,t} = h(a_t, s_{c,t}) \tag{6}$$

This selection function combines beliefs from both models and can be implemented in multiple ways. In this work we fuse the scores with a convex combination $h(a_t, s_{c,t}) = \beta a_t + (1-\beta)s_{c,t}$ and leave other possible architectures for future work. After $h_{c,t}$ is computed, we proceed as before and select top-k instances with the highest $h_{c,t}$ values to get $\mathcal{T}^c$.

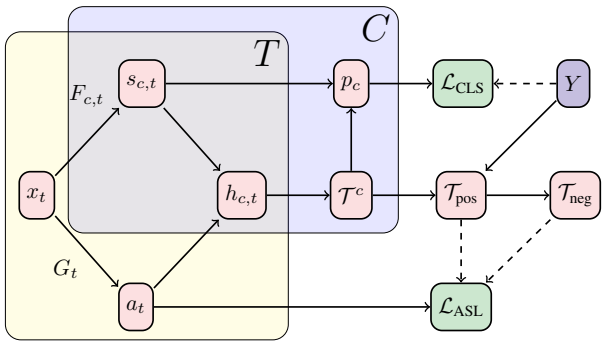
To train the actionness model G, we design a novel task to predict whether a given instance x_t will be in the top-k set for any ground truth class. Since context is highly class dependent, we hypothesize that G can only perform well on this task by learning to capture action characteristics that are ubiquitous across classes. This hypothesis is further motivated by the fact that many leading supervised localization methods first generate class-agnostic proposals and then predict classes for them [17, 18, 16]. High accuracy of these models indicates that the proposal network is able to learn the general actionness property independent of the class, and we aim to do the same here. We first partition instances into positive and negative sets:

$$\mathcal{T}_{\text{pos}} = \bigcup_{c\in Y}\mathcal{T}^c \tag{7}$$

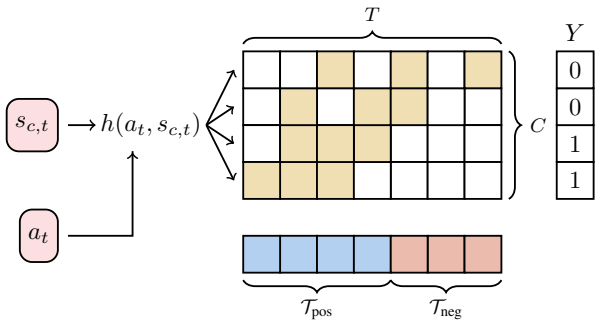
$$\mathcal{T}_{\text{neg}} = \{1, ..., T\}\backslash\mathcal{T}_{\text{pos}} \tag{8}$$

where positive set $\mathcal{T}_{\text{pos}}$ contains the union of all instances that appear in the top-k for the ground truth classes Y, and negative set $\mathcal{T}_{\text{neg}}$ has all other instances. We then train G to predict whether each instance is in the positive or negative set. In our model the classifier and actionness networks are tied by the instance selection function. Empirically, we observe that during training as classification accuracy improves better instances get selected into positive and negative sets. This improves the actionness model which translates to better top-k instance selection for the classifier, leading to further improvement in classification accuracy. The two models are thus complementary to each other, and we show that *both* classification and localization accuracy improve when the actionness network is added.

Since our target positive and negative sets contain both context and action instances, the binary inclusion labels can be noisy. This is particularly the case early in training when classification accuracy is poor and top selected instances are not accurate. Traditional cross entropy classification loss assigns a large penalty when prediction de-



(a) Diagram of the proposed approach.



(b) Toy example with $C = 4, T = 7, k = 3$.

Figure 2: ASL model architecture and toy example. (a) For every instance x_t the classifier $F_{c,t}$ predicts class activation $s_{c,t}$, and actionness model G_t predicts actionness score a_t. Class activation and actionness are combined with the instance selection function h to get instance selection probability $h_{c,t} = h(a_t, s_{c,t})$. Top-$k$ instances with the highest selection probabilities are then added to $\mathcal{T}^c$ and aggregated together to generate class prediction p_c for the video. Finally, the union of top-k instances across ground-truth classes Y is used to generate target sets $\mathcal{T}_{\text{pos}}$ and $\mathcal{T}_{\text{neg}}$ for the actionness model. (b) Toy example illustrates how target sets $\mathcal{T}_{\text{pos}}$ and $\mathcal{T}_{\text{neg}}$ are computed. The video has $T = 7$ instances, $C = 4$ classes and $k = 3$. For each class we select top-3 instances with the highest action selection probabilities $h(a_t, s_{c,t})$ indicated by yellow cells. Taking union of selected instances across ground truth classes ($c \in Y$) we get $\mathcal{T}_{\text{pos}}$ shown in blue. All other instances form $\mathcal{T}_{\text{neg}}$ shown in red.

viates significantly from the ground truth. This is a desirable property when labels are clean, enabling the model to converge quickly [38]. However, recent work shows that cross entropy leads to poor performance under noisy labels, where the high penalty can force the model to overfit to noise [8, 38]. To address this problem the generalized cross entropy loss has been proposed that softens the penalty in regions of high disagreement [38]. We adopt this loss here to improve the generalization of the actionness model:

$$\mathcal{L}_{\text{ASL}} = \frac{1}{|\mathcal{T}_{\text{pos}}|} \sum_{t \in \mathcal{T}_{\text{pos}}} \frac{1 - (a_t)^q}{q} + \frac{1}{|\mathcal{T}_{\text{neg}}|} \sum_{t \in \mathcal{T}_{\text{neg}}} \frac{1 - (1 - a_t)^q}{q} \tag{9}$$

where $0 < q \leq 1$ controls the noise tolerance. $\mathcal{L}_{\text{ASL}}$ is based on the negative Box-Cox transform [2], and approaches mean absolute error when q is close to 1 which is more tolerant to deviations from ground truth. On the other hand, as q approaches 0, $\mathcal{L}_{\text{ASL}}$ behaves similarly to the cross entropy loss with stronger penalties. By appropriately setting q we can control model sensitivity to noise and improve robustness. During training we optimize both classification and ASL losses simultaneously $\mathcal{L} = \mathcal{L}_{\text{CLS}} + \mathcal{L}_{\text{ASL}}$ and backpropagate the gradients through both classifier and actionness networks.

The proposed ASL architecture is summarized in Figure 2(a). Figure 2(b) also shows a toy example that illustrates how positive and negative sets $\mathcal{T}_{\text{pos}}$ and $\mathcal{T}_{\text{neg}}$ are computed. The video has $T = 7$ instances and $C = 4$ classes, two of which are in the ground truth $Y = \{3, 4\}$. Moreover, $k = 3$ so for each class top-3 instances with the highest instances selection probabilities $h(a_t, s_{c,t})$ are selected, indicated by yellow cells. Union of instances selected for the ground truth classes form $\mathcal{T}_{\text{pos}} = \{x_1, x_2, x_3, x_4\}$ shown in red, and all other instances form $\mathcal{T}_{\text{neg}} = \{x_5, x_6, x_7\}$ shown in blue. To successfully predict instances in each list, the actionness model must find commonalities between all instances in $\mathcal{T}_{\text{pos}}$ and distinguish them from $\mathcal{T}_{\text{neg}}$. As we demonstrate in the experimental section this commonality is the presence of actionness which significantly aids the localization task.

After training, we use the instance selection probabilities $h_{c,t}$ to localize actions in test videos. Given a test video with T' instances, we run it through our model to get the corresponding instance selection probability sequence $h_{c,1}, ..., h_{c,T'}$. We then follow recent work [23, 14, 28] and apply multiple thresholds $0 < \alpha < 1$. All instances where selection probability is above the threshold $h_{c,t} > \alpha$ are considered selected, and we take all consecutive sequences as proposal candidates. Repeating this process for each threshold, we obtain a set of proposals for each class. We then apply non-maximal suppression to eliminate overlapping and similar proposals and generate the final localization predictions.

4. Experiments

We conduct extensive experiments on two popular weakly supervised temporal localization datasets containing untrimmed videos: THUMOS-14 [11] and ActivityNet-1.2 [3]. THUMOS-14 contains 200 training videos with 20 action classes and 212 test videos. ActivityNet-1.2 contains 4,819 training and 2,383 test videos with 100 action classes.

Table 1: THUMOS-14 results. mAP is the mean of AP@IoU scores across thresholds $\{0.1, 0.2, ..., 0.9\}$. Ablation results are shown at the bottom where ASL-$s_{c,t}$ and ASL-a_t use class probability $s_{c,t}$ and actionness probability a_t respectively to localize, and ASL-BCE trains the actionness network G with the binary cross entropy loss.

Approach	AP@IoU					mAP
	0.1	0.3	0.5	0.7	0.9	
CMCS [19]	57.4	41.2	23.1	7.0	-	-
MAAN [36]	59.8	41.1	20.3	6.9	0.2	24.9
3C-Net [23]	56.8	40.9	24.6	7.7	-	-
BaSNet [14]	58.2	44.6	27.0	10.4	0.5	27.9
BM [25]	60.4	46.6	26.8	9.0	0.4	28.6
DGAM [28]	60.0	46.8	28.8	11.4	0.4	29.2
ACL [10]	-	46.9	30.1	10.4	-	-
TSCN [37]	63.4	47.8	28.7	10.2	0.7	22.9
EM-MIL [22]	59.1	45.5	30.5	**16.4**	-	-
ASL (ours)	**67.0**	**51.8**	**31.1**	11.4	**0.7**	**32.2**
Ablation						
ASL-$s_{c,t}$	56.9	40.5	19.7	6.0	0.4	24.0
ASL-a_t	55.9	40.3	20.6	6.8	0.4	30.4
ASL-BCE	66.4	50.5	30.5	10.9	0.7	31.6

Both datasets have videos that vary significantly in length from a few seconds to over 25 minutes. This makes the problem challenging since the model has to perform well on both long and short action sequences. For all experiments, we only use video-level class labels during training. To make the comparison fair, we follow the same experimental setup used in literature [26, 23, 14, 28], including data splits, evaluation metrics and input features. For all experiments, we report average precision (AP) at the different intersection over union (IoU) thresholds between predicted and ground truth localizations. For brevity, we show selected thresholds in the results table for both datasets. The mAP is computed with IoU thresholds between 0.1 to 0.9 with increments of 0.1 on THUMOS-14 and between 0.5 to 0.95 with increments of 0.05 on ActivityNet-1.2 to stay consistent with previous work. Full results on all thresholds used for computing mAP are found in the supplementary material.

Implementation Details We generate instance input features x_t following the same pipeline as recent leading approaches [23, 14, 28]. The I3D network [6] pre-trained on the Kinetics dataset [12] is applied on each sub-sequence of 16 consecutive frames with RGB and TVL1 flow [33] inputs to extract 2048-dimensional feature representation by spatiotemporally pooling the Mixed5c layer. Linear interpolation across time is then applied for both datasets. To make the comparison fair we adopt the same base classifier for F as in [14] with 512 hidden units and ReLU activations. Similarly, the actionness network G is fully connected with 512 hidden units. Both networks are applied across time to every instance and operate similarly to convolutional filters with kernel size 1. We set noise tolerance $q = 0.7$ for both datasets, and use previously reported instance selection parameters $k = T/8$ for THUMOS-14 and ActivityNet-1.2 [23, 14]. We set β to be 0.5 for the instance selection function $h(a_t, s_{c,t})$. ASL is trained using the ADAM optimizer [13] with batch size 16, learning rate 1e-4 and weight decay 1e-4 until convergence. During inference, we use ten localization thresholds α from 0 to 1 to generate localization proposals. We then compute final predictions by applying non-maximum suppression to eliminate overlapping and similar proposals.

Table 2: ActivityNet-1.2 results. mAP is the mean of AP@IoU scores across thresholds $\{0.5, 0.55, ..., 0.95\}$.

Approach	AP@IoU					mAP
	0.5	0.6	0.7	0.8	0.9	
TSM [35]	28.3	23.6	18.9	14.0	7.5	17.1
3CNet [23]	35.4	-	22.9	-	8.5	21.1
CMCS [19]	36.8	-	-	-	-	
CleanNet [20]	37.1	29.9	23.4	17.2	9.2	21.6
BaSNet [14]	38.5	-	-	-	-	24.3
DGAM [28]	**41.0**	33.5	26.9	19.8	10.8	24.4
TSCN [37]	37.6	-	-	-	-	23.6
EM-MIL [22]	37.4	-	23.1	-	2.0	20.3
ASL (ours)	40.2	**34.6**	**29.4**	**22.5**	**12.1**	**25.8**

We compare our approach with an extensive set of leading recent baselines: TSM [35], CMCS [19], MAAN [36], 3C-Net [23], CleanNet [20], BaSNet [14], BM [25], DGAM [28], TSCN [37] and EM-MIL [22]. Details for each baseline can be found in the related work section, and we directly use the results reported by the respective authors.

Results Table 1 summarizes temporal localization results on the THUMOS-14 dataset. Our approach improves over the prior art by a significant margin on all IoU thresholds except 0.7, with a 10.3% relative gain in mAP over the best baseline. A similar pattern can be observed from the ActivityNet-1.2 results summarized in Table 2. We can see that ASL improves over every baseline on all IoU thresholds except 0.5, with a 5.7% relative gain in mAP. These results indicate that the proposed action selection learning framework is highly effective for the weakly supervised temporal localization task. Figure 3 further breaks down THUMOS-14 performance by class. The top four classes with the largest relative improvement are highlighted in blue. The most improved classes have more than 100% relative gain. We believe this is due to the wide range of context settings present in the dataset for these classes, making class-agnostic action learning more effective for separating context.

Ablation To demonstrate the importance of actionness, we conduct ablation study on the THUMOS-14 dataset shown at the bottom of Table 1. Here, ASL-$s_{c,t}$ uses class probability $h_{c,t} = s_{c,t}$ and ASL-a_t uses actionness proba-

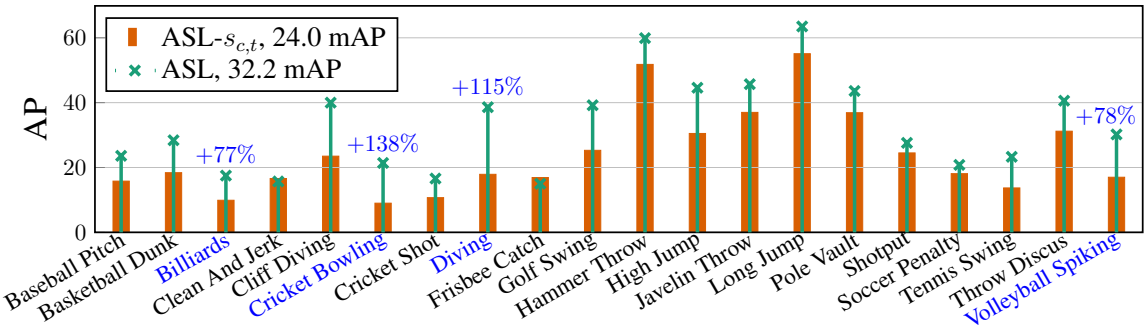
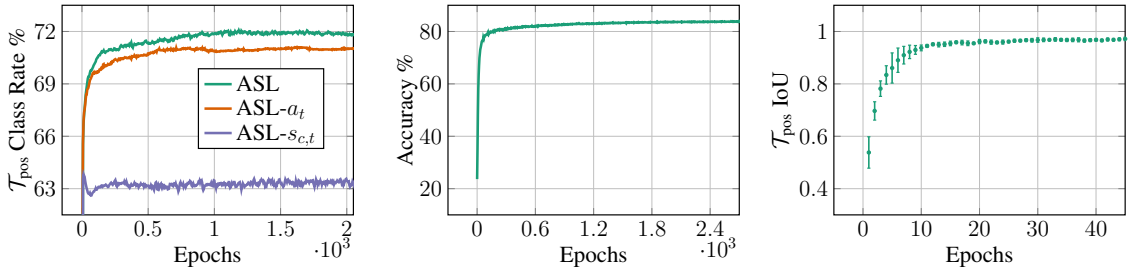


Figure 3: THUMOS-14 per-class AP results, four largest relative improvements are highlighted in blue.



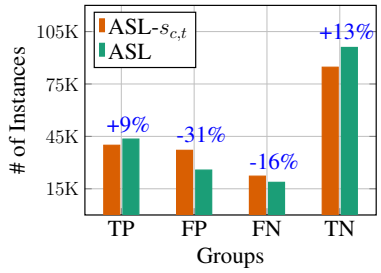
(a) (b) (c)

Figure 4: THUMOS-14 $\mathcal{T}_{\text{pos}}$ analysis. (a) Fraction of instances in $\mathcal{T}_{\text{pos}}$ that contain ground truth action from any target class. (b) Validation ASL accuracy at predicting which instances will be in $\mathcal{T}_{\text{pos}}$. (c) Averaged across training videos IoU of $\mathcal{T}_{\text{pos}}$ sets between consecutive epochs, error bars show one standard deviation.

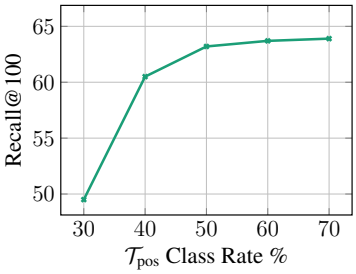
bility $h_{c,t} = a_t$ for localization proposals during inference. The classification for ASL-a_t is done at the video level by taking the class with the highest probability and assigning it to every localization. Finally, ASL-BCE trains the actionness network G with the binary cross entropy loss instead of the generalized $\mathcal{L}_{\text{ASL}}$ loss in Equation 9. We see that incorporating actionness in the full ASL model relatively outperforms the classification-only ASL-$s_{c,t}$ approach by over 36% in mAP. Moreover, ASL-a_t has very strong performance and is competitive with prior state-of-the-art results even though a_t on its own has no explicit class information. This demonstrates that the actionness network G is able to successfully capture the general class-agnostic concept of action through our top-k instance prediction task. Once captured, this property can be effectively used to identify regions within each video where the action occurs independently of the class. We note here that attention models commonly used in prior work, have not been shown to be capable of localizing actions on their own. The full ASL model further improves performance of ASL-a_t by 6% indicating that classifier and actionness networks capture complementary information. Finally, using binary cross entropy instead of the noise tolerant $\mathcal{L}_{\text{ASL}}$ loss hurts performance. $\mathcal{L}_{\text{ASL}}$ becomes equivalent to binary cross entropy in the limit as $q \to 0$ [38]. In all our experiments we found that much higher values of q such as 0.7 produced better performance on both datasets, indicating that cross entropy is indeed not adequate here due to the high degree of noise in the target labels particularly at the beginning of training.

Actionness Learning The main idea behind ASL is that actionness can be captured by predicting top-k membership for each instance. In this section, we analyze this learning task in detail. Figure 4 shows various properties of the $\mathcal{T}_{\text{pos}}$ set throughout training. In Figure 4(a) we plot the fraction of instances in $\mathcal{T}_{\text{pos}}$ that contain ground truth action from any target class over training epochs. For comparison, we also plot this fraction for ASL-$s_{c,t}$ and ASL-a_t where top-k instances are chosen according to class $s_{c,t}$ and actionness a_t probabilities respectively. We observe that without the actionness model, ASL-$s_{c,t}$ hovers around 63% whereas for ASL it steadily increases to over 72%. Furthermore, ASL-a_t reaches a much higher fraction of over 70% compared to ASL-$s_{c,t}$, capturing a significantly larger portion of instances with action. This again indicates that the actionness network is better at identifying action instances than the classifier.

Figure 4(b) shows the validation accuracy of the actionness model G in predicting which instances are in the $\mathcal{T}_{\text{pos}}$ set. Despite the fact that $\mathcal{T}_{\text{pos}}$ is a moving target that can change with each iteration, the prediction accuracy remains stable and gradually improves throughout learning reaching over 84%. The model is thus able to reach an equilibrium between the two networks and no divergence is observed. Furthermore, Figure 4(c) shows the intersection over union



(a)



(b)

Method	mAP	AP@0.5
F and G	32.2	31.1
F then G	26.8	25.7
F:30 G:30	30.3	28.8
F:20 G:20	30.7	29.6
F:10 G:10	30.4	28.9
F:5 G:5	30.9	29.6
F:1 G:1	31.1	30.0

(c)

Figure 5: THUMOS-14 error and ablation analysis. (a) Error analysis across all test instances by type. TP, FP, TN, FN indicate true positives, false positives, true negatives and false negatives respectively. (b) Class-agnostic localization Recall@100 using G when it is trained from scratch on $\mathcal{T}_{\text{pos}}$ with different proportion of instances that contain action. (c) Effect of training schedules in ASL. F and G is joint ASL training, F then G is sequential training of F followed by G, F:n G:m is an alternating schedule where F is updated for n epochs followed by G for m epochs.

(IoU) between the $\mathcal{T}_{\text{pos}}$ sets from consecutive epochs during training. A higher IoU indicates a larger overlap between consecutive $\mathcal{T}_{\text{pos}}$ sets which in turn makes targets for G more stable and easier to learn. We observe that the initial IoU starts around 0.5 and rapidly approaches 1 as training progresses. Furthermore, the variance in IoU across training videos decreases throughout training so $\mathcal{T}_{\text{pos}}$ sets stabilize after the first few epochs. These results indicate that the top-k selection remains consistent for the majority of training epochs, and G is able to successfully learn these targets; we observe this pattern in all our experiments.

Figure 5a breaks down the predictions made by ASL by error type. We use all test instances and show the total number of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN). Note that we treat this as a binary problem and consider an instance as true positive if it contains an action and is selected for localization by ASL. In this setting, FP and FN represent context and actionness errors respectively, TP and TN are correctly predicted instances. We compare ASL with the classifier-only ASL-$s_{c,t}$ model and show the relative increase/decrease for each category in blue. Figure 5a shows that ASL improves each category predicting more instances correctly and making fewer mistakes. Specifically, ASL reduces context and actionness errors by 31% and 16% respectively.

In Figure 4 we showed that ASL can capture actionness because a significant proportion of instances in $\mathcal{T}_{\text{pos}}$ sets contain actions, and these sets remain stable throughout training. Moreover, even though not all instances in $\mathcal{T}_{\text{pos}}$ contain actions, our noise tolerant loss is robust and can still perform well when a portion of labels is incorrect. Here, we further investigate the degree of noise that can be tolerated in this setting. Specifically, we evaluate the ability of G to capture actionness when it is trained on $\mathcal{T}_{\text{pos}}$ with different proportions of action instances. Throughout training, we sample instances from $\mathcal{T}_{\text{pos}}$ sets to lower the fraction of instances that contain actions. This simulates a challenging learning environment where G has to learn from increasingly noisier labels. Figure 5b shows these results on the THUMOS-14 dataset where we reduce the fraction of instances containing actions in $\mathcal{T}_{\text{pos}}$ (Class Rate) from 70% to 30%. For comparison, the actual class rate on this dataset reaches around 72% (see Figure 4(a)). To more directly evaluate the impact of this setting we compute the instance-level Recall@100 using only predictions from G. Recall@100 is computed by measuring the fraction of instances that contain any action amongst the top-100 instances predicted by G. We can see that Recall@100 for the noise-tolerant ASL setting remains relatively stable between 50% and 70% but starts to drop significantly below 40% class rate. These results suggest that the top-k selection strategy can tolerate a high degree of noise with up to 50% of incorrect labels. However, this also implies that the classifier needs to be sufficiently accurate in the top-k selection for ASL to work.

Throughout the training, we simultaneously update both F and G. We discussed that this results in moving targets for G where instances in $\mathcal{T}_{\text{pos}}$ change as the classifier is updated. Alternative training strategies are explored in Figure 5c. We experiment with first training F to convergence and then G (F then G), and alternating between training F and G. We denote these alternating schedules by F:n G:m to indicate training F for n epochs followed by training G for m epochs and repeating. We observe that training F then G results in the lowest performance, since the model cannot adjust the classifier to work better with the actionness model. Alternating between F and G updates improves performance but still lags behind joint training. This corroborates our intuition that the classifier and actionness models complement each other in ASL and should be trained together.

We further show DETAD [1] analysis on THUMOS-14 dataset in Figure 7. Here, the top row shows false positive analysis (context error) and false negative analysis (action-

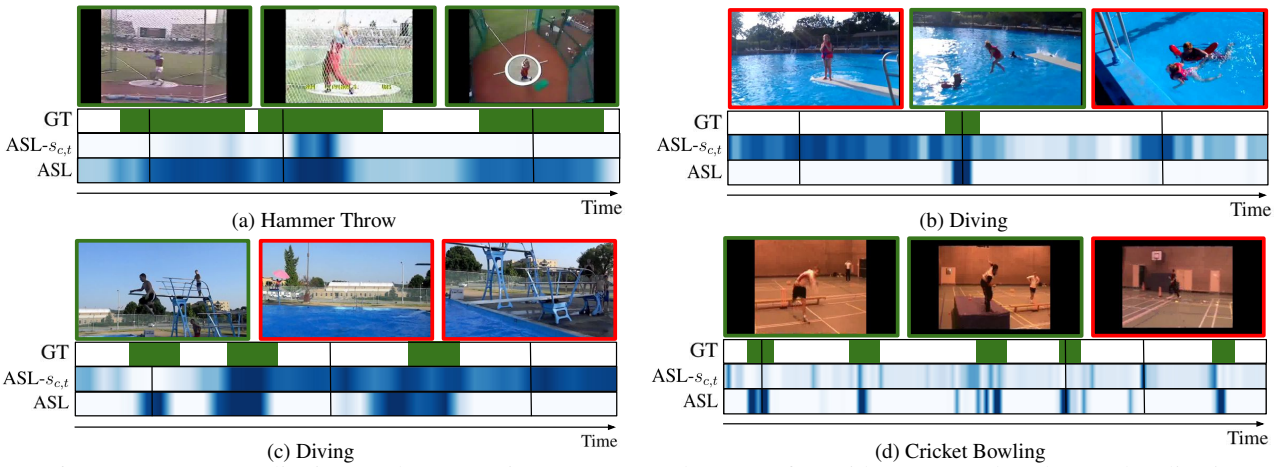


Figure 6: THUMOS-14 qualitative results comparing ASL-$s_{c,t}$ and ASL on four videos. Ground truth (GT) localizations are indicated with green segments, and we show sample frames from action (green) and context (red) instances. Predictions are shown in blue where darker color indicates higher confidence.

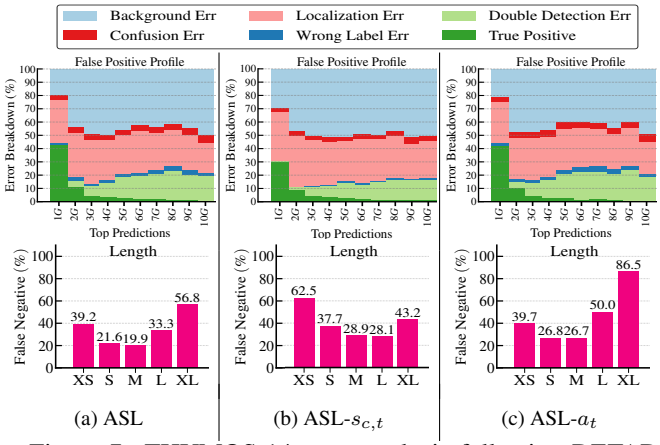


Figure 7: THUMOS-14 error analysis following DETAD [1]. Top row breaks down false positives errors while bottom row shows false negatives by segment lengths.

ness error) is shown in the bottom row. The false positive profiles show that ASL-a_t significantly reduces the background (41.7 vs 47.7) and localization (28.3 vs 32.5) errors compared to ASL-$s_{c,t}$, further demonstrating that G is able to learn actionness concepts that the classifier fails to capture. On the other hand, we observe higher double detection (14.8 vs 9.6) and wrong label (2.9 vs 0.9) errors in ASL-a_t since it lacks class information. This corroborates our hypothesis that both models are needed to maximize localization accuracy. In the top predictions 1G, nearly all of the reduction in background error translates to more true positives in both ASL and ASL-a_t compared to ASL-$s_{c,t}$. On the false negatives, ASL-a_t improves on shorter and more frequent action segments and complements ASL-$s_{c,t}$ which captures longer and more infrequent action segments better.

Qualitative Results Figure 6 shows qualitative results for ASL-$s_{c,t}$ and ASL models. Ground truth segments are shown in green, and model predictions are shown in blue with darker colors indicating higher confidence. In figure 6(a) for a video containing the "Hammer throw" action, the ASL-$s_{c,t}$ localization alone is focusing on one very specific region of the video which likely contains the easiest instances (actionness error) to predict the video class. ASL spreads the localization predictions and correctly identifies all regions of actions. The opposite pattern is observed in Figures 6(b) and 6(c) which show the "Diving" action class. ASL-$s_{c,t}$ activations are high on many context instances (context error) as they all contain highly informative scenes for the target "Diving" class. ASL, however, focuses on the regions that contain the action. Lastly, Figure 6(d) shows "Cricket Bowling" action in an uncommon indoor setting. Here, ASL-$s_{c,t}$ has difficulty recognizing the action and most activations have low confidence. The actionness model is able to identify the action of throwing as shown in the first two sample frames. Predictions of ASL are significantly more confident and identifies all of the action segments.

5. Conclusion

We propose the Action Selection Learning (ASL) approach for weakly supervised video localization. ASL incorporates a class-agnostic actionness network that learns a general concept of action independent of the class. We train the actionness network with a novel prediction task by classifying which instances will be selected in the top-k set by the classifier. Once trained, this network is highly effective on its own and can accurately localize actions with minimal class information from the classifier. Empirically, ASL demonstrates superior accuracy, outperforming leading recent benchmarks by a significant margin. Future work includes further investigation into actionness and its generalization to other related video domains.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, 2018.

[2] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.

[3] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[5] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, 2017.

[7] Wei Chen, Caiming Xiong, Ran Xu, and Jason J Corso. Actionness ranking with lattice conditional ordinal random fields. In *CVPR*, 2014.

[8] Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[10] Guoqiang Gong, Xinghan Wang, Yadong Mu, and Qi Tian. Learning temporal co-attention models for unsupervised video action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[11] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014.

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, 2020.

[15] Nannan Li, Dan Xu, Zhenqiang Ying, Zhihao Li, and Ge Li. Searching action proposals via spatial actionness estimation and temporal path inference and tracking. In *ACCV*, 2016.

[16] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-matching network for temporal action proposal generation. In *ICCV*, 2019.

[17] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *ACM Multimedia*, 2017.

[18] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018.

[19] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*, 2019.

[20] Ziyi Liu, Le Wang, Qilin Zhang, Zhanning Gao, Zhenxing Niu, Nanning Zheng, and Gang Hua. Weakly supervised temporal action localization through contrast based evaluation networks. In *ICCV*, 2019.

[21] Ye Luo, Loong-Fah Cheong, and An Tran. Actionness-assisted recognition of actions. In *ICCV*, 2015.

[22] Zhekun Luo, Devin Guillory, Baifeng Shi, Wei Ke, Fang Wan, Trevor Darrell, and Huijuan Xu. Weakly-supervised action localization with expectation-maximization multi-instance learning. *arXiv preprint arXiv:2004.00163*, 2020.

[23] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *CVPR*, 2019.

[24] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, 2018.

[25] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *ICCV*, 2019.

[26] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, 2018.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[28] Baifeng Shi, Qi Dai, Yadong Mu, and Jingdong Wang. Weakly-supervised action localization by generative attention modeling. *arXiv preprint arXiv:2003.12424*, 2020.

[29] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 2018.

[30] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017.

[31] Limin Wang, Yu Qiao, Xiaoou Tang, and Luc Van Gool. Actionness estimation using hybrid fully convolutional networks. In *CVPR*, 2016.

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

[33] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l 1 optical flow. In *Statistical and geometrical approaches to visual motion analysis*. 2009.

[34] Gang Yu and Junsong Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015.

[35] Tan Yu, Zhou Ren, Yuncheng Li, Enxu Yan, Ning Xu, and Junsong Yuan. Temporal structure mining for weakly supervised action detection. In *ICCV*, 2019.

[36] Yuan Yuan, Yueming Lyu, Xi Shen, Ivor W Tsang, and Dit-Yan Yeung. Marginalized average attentional net-

work for weakly-supervised learning. *arXiv preprint arXiv:1905.08586*, 2019.

[37] Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, Junsong Yuan, and Gang Hua. Two-stream consensus network for weakly-supervised temporal action localization. In *European Conference on Computer Vision*, pages 37–54. Springer, 2020.

[38] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.

[39] Hang Zhao, Zhicheng Yan, Heng Wang, Lorenzo Torresani, and Antonio Torralba. Slac: A sparsely labeled dataset for action classification and localization. *arXiv preprint arXiv:1712.09374*, 2017.