# Actionness Inconsistency-Guided Contrastive Learning for Weakly-Supervised Temporal Action Localization

**Zhilin Li, Zilei Wang*, Qinying Liu**

University of Science and Technology of China, Hefei, China
lizhilin@mail.ustc.edu.cn, zlwang@ustc.edu.cn, lydyc@mail.ustc.edu.cn

## Abstract

Weakly-supervised temporal action localization (WTAL) aims to detect action instances given only video-level labels. To address the challenge, recent methods commonly employ a two-branch framework, consisting of a class-aware branch and a class-agnostic branch. In principle, the two branches are supposed to produce the same actionness activation. However, we observe that there are actually many inconsistent activation regions. These inconsistent regions usually contain some challenging segments whose semantic information (action or background) is ambiguous. In this work, we propose a novel Actionness Inconsistency-guided Contrastive Learning (AICL) method which utilizes the consistent segments to boost the representation learning of the inconsistent segments. Specifically, we first define the consistent and inconsistent segments by comparing the predictions of two branches and then construct positive and negative pairs between consistent segments and inconsistent segments for contrastive learning. In addition, to avoid the trivial case where there is no consistent sample, we introduce an action consistency constraint to control the difference between the two branches. We conduct extensive experiments on THUMOS14, ActivityNet v1.2, and ActivityNet v1.3 datasets, and the results show the effectiveness of AICL with state-of-the-art performance. Our code is available at https://github.com/lizhilin-ustc/AAAI2023-AICL.

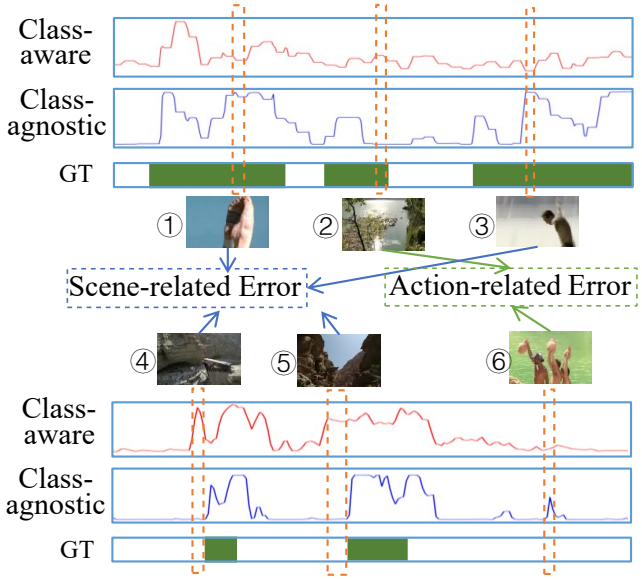


Figure 1: Illustration of the "CliffDiving" action scores for the class-agnostic and class-aware branches. Curves represent segment-wise activation scores.

## Introduction

Temporal action localization is a task to localize the start and end timestamps of action instances and recognize their categories. In recent years, many works (Nawhal and Mori 2021; Zhao et al. 2017; Zhai et al. 2020; Zhao et al. 2022) put effort into a fully-supervised manner and gain great achievements. However, these fully-supervised methods require extensive manual segment-level annotations. Different from the fully-supervised method, the weakly-supervised method only needs video-level labels for training, that is, each video only needs to give the action category that appears in it. Meanwhile, weakly-supervised methods can avoid noise in frame-level annotations. Most of the existing methods use a multi-instance learning mechanism, where a video-level classifier is trained using the video-level annotations of action categories. Using the classifier, we can obtain a sequence of class logits or predictions, *i.e.*, class activation sequence (CAS). However, the learned CAS is usually noisy due to the lack of frame-level annotations.

To this end, recent methods (Lee, Uh, and Byun 2020; Huang, Wang, and Li 2021; Ma et al. 2021; Qu et al. 2021) typically employ a two-branch structure, which is comprised by a class-aware branch and a class-agnostic branch. The class-aware branch uses multi-instance learning and video-level labels to learn category-specific T-CAS, while the class-agnostic branch focuses on the general action in the video. By digging into these two branches, we find that the two branches do not always activate the same regions. To be specific, there are mainly two types of cases that account for the differences. The first type is the error related to scene information. For example, in Figure 1, ④ and ⑤ are non-action regions but contain scene information (such

*Corresponding author

as "cliffs" in Figure 1) for specific action class. For these two cases, the class-aware branch generates wrong action activation (i.e. high action score), while the class-agnostic branch will not as there is no obvious action cue. ① and ③ are two action regions without scene information, so the class-aware branch will not generate the corresponding action activation. But yet, the class-agnostic branch does. The second type is the error related to action information. For example, the human body in ② is small and there is no significant body movement, thus the class-agnostic branch will not produce action activation. However, the scene information in ② is very conspicuous, so the class-aware branch produces the corresponding action activation. In ⑥, there is a "cheering" action unrelated to the label category, the class-agnostic branch produces false action activation since it can identify general action information and is unable to distinguish action categories. Obviously, for segments that are activated simultaneously in both branches (named consistent segments), the model can easily determine whether they are actions or backgrounds. For ambiguous segments that do not have simultaneous activations in both branches (named inconsistent segments), the model cannot perform accurate action/background classification, which is harmful to the final performance. It comes into our mind: is it possible to utilize the accurate consistent segments to refine the inconsistent segments?

In recent years, contrastive learning(Chen et al. 2020; He et al. 2020) has been proven to be an effective way to learn feature representations. The goal of contrastive learning is to learn a feature encoder that allows anchors and positive samples to learn similar representations, and negative samples to learn more different representations. Inspired by these works, we propose Actionness Inconsistency-guided Contrastive Learning(AICL), which aims to use contrastive learning as a tool to improve the learning of inconsistent segments. Specifically, we first define the inconsistent segments with higher/lower action scores as inconsistent action/background segments. For the inconsistent action segments, we enforce their representation to be as close to the representation of the consistent action segments as possible, and away from that of the consistent background segments; for inconsistent background segments, we adopt the opposite strategy. By this means, the representation of inconsistent segments is refined with the help of consistent segments.

Although contrastive learning is introduced, we observe that there is still an issue. To be specific, the difference between the action predictions of the two branches will gradually increase as the model training progresses, resulting in the inability to select enough consistent action/background segments for contrastive learning. In order to maintain contrastive learning, we impose a two-branch action consistency constraint that collaboratively trains these two branches to reduce the inconsistency between the two branches.

To summarize, our contribution is three-fold:

- We highlight that there exists inconsistency between the class-aware branch and the class-agnostic branch, and propose a WTAL framework named AICL, which leverages contrastive learning to make the representation of inconsistent segments more discriminative.
- We propose an action consistency constraint to reduce the difference between the class-aware branch and the class-agnostic branch, so as to maintain enough consistent segments for contrastive learning.
- We conduct extensive experiments on three public benchmarks, where our proposed method achieves state-of-the-art results.

## Related Work

**Contrastive Representation Learning.** Contrastive Representation Learning (Hadsell, Chopra, and LeCun 2006; Chen et al. 2020; He et al. 2020; Chen and He 2021) focuses on learning common features between instances of the same class and distinguishing differences between instances of different classes. In recent years, contrastive learning has made impressive progress in unsupervised representation learning. It uses data internal patterns to learn an embedding space where associated signals are brought together while unassociated ones are distinguished via Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen 2010). SimCLR (Chen et al. 2020) randomly samples a part of the samples as a batch, performs two different data enhancements on the samples in a batch, and hopes that the same sample with different data enhancements is as close as possible and far away from other samples. MoCo (He et al. 2020) uses a momentum updated memory bank of old negative representations to get rid of the batch size restriction and enable the consistent use of negative samples.

**Weakly-Supervised Temporal Action Localization.** In order to save the cost of labeling and avoid labeling noise, weakly-supervised temporal action localization(WTAL) is proposed. Compared to fully-supervised temporal action localization, which requires precise annotation of each action instance, weakly-supervised temporal action localization methods only require video-level action class labels during training. At present, in the field of weak supervision, there are two main methods: one(Islam, Long, and Radke 2021; Lee, Uh, and Byun 2020; Liu et al. 2021b; Zhang et al. 2021; Ma et al. 2021) first obtains frame-level action scores, i.e. the class activation sequence CAS, and then introduce a top-k mechanism to construct video-level classification scores. This method tackles this problem using the multiple-instance learning (MIL) framework(Carbonneau et al. 2018). Another(Nguyen et al. 2018; Yu et al. 2019; Nguyen, Ramanan, and Fowlkes 2019; Min and Corso 2020) formulates this task as an action recognition problem and introduces a foreground-background separation attention branch to construct video-level features, then apply an action classifier to recognize videos.

**Contrastive Learning in Weakly-Supervised Temporal Action Localization.** To enhance representation learning, many weakly-supervised methods introduce contrastive learning. CoLA (Zhang et al. 2021) introduces the segment contrastive estimation to the WTAL task. They hope to improve the representation of hard segments through contrastive learning. FTCL (Gao, Chen, and Xu 2022) argues that learning by contextually comparing sequence-to-

sequence distinctions offers an essential inductive bias in the WTAL and helps identify coherent action instances. DCC (Li et al. 2022) proposes a novel denoised cross-video contrastive algorithm, aiming to reduce the impact of wrong division of positive/negative sample pairs.

# Method

## Problem Definition

In this paper, we define $V = \{v_i\}_{i=1}^{L}$ as an untrimmed video with $L$ frames. It is annotated with its video-level action label $\{y_c\}_{c=1}^{C}$, where $y_c$ is a binary vector indicating whether the action exists. $C$ is the number of action classes in the dataset. In the inference stage, the model predicts all action instances in a video, then outputs a series of action instances $\{c, q, t_s, t_e\}$, where $c$ denotes the predicted action class, $q$ is the confidence score, $t_s$ and $t_e$ represent the start time and end time of the action instances.

## Feature Extraction

Following recent WTAL methods (Zhang et al. 2021; Ma et al. 2021), for a given untrimmed video, we first evenly divide it into a fixed number (denoted by $T$) of segments. Each segment has 16 frames. Then we apply a pre-trained feature extractor (i.e., I3D (Carreira and Zisserman 2017)) to extract segment-level features from both RGB and optical-flow streams. We use matrix tensors $F_R \in \mathbb{R}^{T\times D}$ and $F_F \in \mathbb{R}^{T\times D}$ to represent the RGB and optical flow features of the whole video, respectively, where $D$ means the dimension of the feature vector.

## Class-aware Branch

The class-aware branch learns category-related action information with the given video-level action labels. Specifically, we first concatenate $F_R$ and $F_F$ together in the channel dimension, denoted as $F \in \mathbb{R}^{T\times 2D}$, and then feed $F$ into the feature embedding layer $f$ to get the hybrid feature $E \in \mathbb{R}^{T\times D_e}$:

$$E = f\left(F\right) \tag{1}$$

After that, we obtain the segment-level class activation sequence through the classification layer $cls$:

$$\text{CAS} = cls\left(E\right) \tag{2}$$

Then, following (Zhang et al. 2021), we sum $CAS \in \mathbb{R}^{T\times C}$ along the channel dimension followed by the sigmoid function to obtain a class-aware aggregation and use it to represent the actionness score $A$:

$$A = Sigmoid\left(Sum\left(\text{CAS}\right)\right) \tag{3}$$

**Action Loss.** Multiple instance learning (Carbonneau et al. 2018) is used to train the class-aware branch, where top-k pooling is applied over CAS for each class to make video-level class predictions. In order to take both category-dependent action information and category-independent action information into consideration, we propose to combine the CAS of the class-aware branch and the actionness score of the class-agnostic branch to form a new score sequence $A_{score}$ for top-k selection. That is,

$$A_{score} = \gamma \cdot Softmax(\text{CAS}) + (1-\gamma)(A_R + A_F) \tag{4}$$

where $\gamma$ is a hyperparameter that adjusts the fusion scale, $A_R \in \mathbb{R}^{T\times 1}$ and $A_F \in \mathbb{R}^{T\times 1}$ represent the actionness scores output by the class-agnostic branch, which will be introduced in the next section. In Eq4, we extend the dimensions of $A_R$ and $A_F$ from $T \times 1$ to $T \times C$ by copying the value C times. We denote the set of top-k instances for each class $c$ as $\mathcal{T}_c$. According to the selected indexes of the segments in $\mathcal{T}_c$, we aggregate the corresponding regions of CAS to get the video-level category prediction $p_c$. Finally, the action loss $\mathcal{L}_a$ is calculated in the form of cross-entropy loss with the ground-truth $y_c$:

$$\mathcal{L}_a = -\sum_{c=1}^{C} y_c \log p_c \tag{5}$$

## Class-agnostic Branch

The class-aware branch will be biased by the class-specific contextual information (Ma et al. 2021), which affects the accuracy of action localization. To address this problem, we introduce the class-agnostic branch to learn category-independent action information. Specifically, we feed the RGB and optical flow features into two different embedding layers separately. The RGB embedding layer $f_R$ obtains the embedding $E_R \in \mathbb{R}^{T\times D_e}$ from RGB features $F_R$:

$$E_R = f_R\left(F_R\right) \tag{6}$$

Then the actionness score of RGB stream (denoted by $A_R \in \mathbb{R}^{T\times 1}$) is obtained through a classification layer $cls_R$ and $Sigmoid$ function:

$$A_R = Sigmoid\left(cls_R\left(E_R\right)\right) \tag{7}$$

Optical flow features go through a similar process and we will get the action score of optical flow $A_F$. To fuse RGB and optical flow information, we compute the final actionness score $A_{ca}$ as follows:

$$A_{ca} = \frac{(A_F + A_R)}{2} \tag{8}$$

**Class-agnostic Loss.** To obtain the supervision signal for training the class-agnostic branch, we adopt a simple pseudo-label generation method proposed in ASL(Ma et al. 2021), which selects the K segments with the highest scores in the categories presented in the video-level labels as the action segments (denoted by $\mathcal{T}^a$), and the rest are background segments (denoted by $\mathcal{T}^b$). Due to the low quality of pseudo-labels, we use generalized cross-entropy loss (Zhang and Sabuncu 2018) to calculate the action-background binary classification loss of RGB $\mathcal{L}_{ca_R}$:

$$\mathcal{L}_{ca_R} = \frac{1}{|\mathcal{T}^{\text{a}}|}\sum_{t\in\mathcal{T}^{\text{a}}}\frac{1-(A_{R;t})^q}{q} + \frac{1}{|\mathcal{T}^{\text{b}}|}\sum_{t\in\mathcal{T}^{\text{b}}}\frac{1-(1-A_{R;t})^q}{q} \tag{9}$$

Like $\mathcal{L}_{ca_R}$, we can get action-background binary classification loss for the optical flow stream (denoted by $\mathcal{L}_{ca_F}$), thus the overall class-agnostic loss (denoted by $\mathcal{L}_{ca}$) can be computed as follows:

$$\mathcal{L}_{ca} = \mathcal{L}_{ca_R} + \mathcal{L}_{ca_F} \tag{10}$$

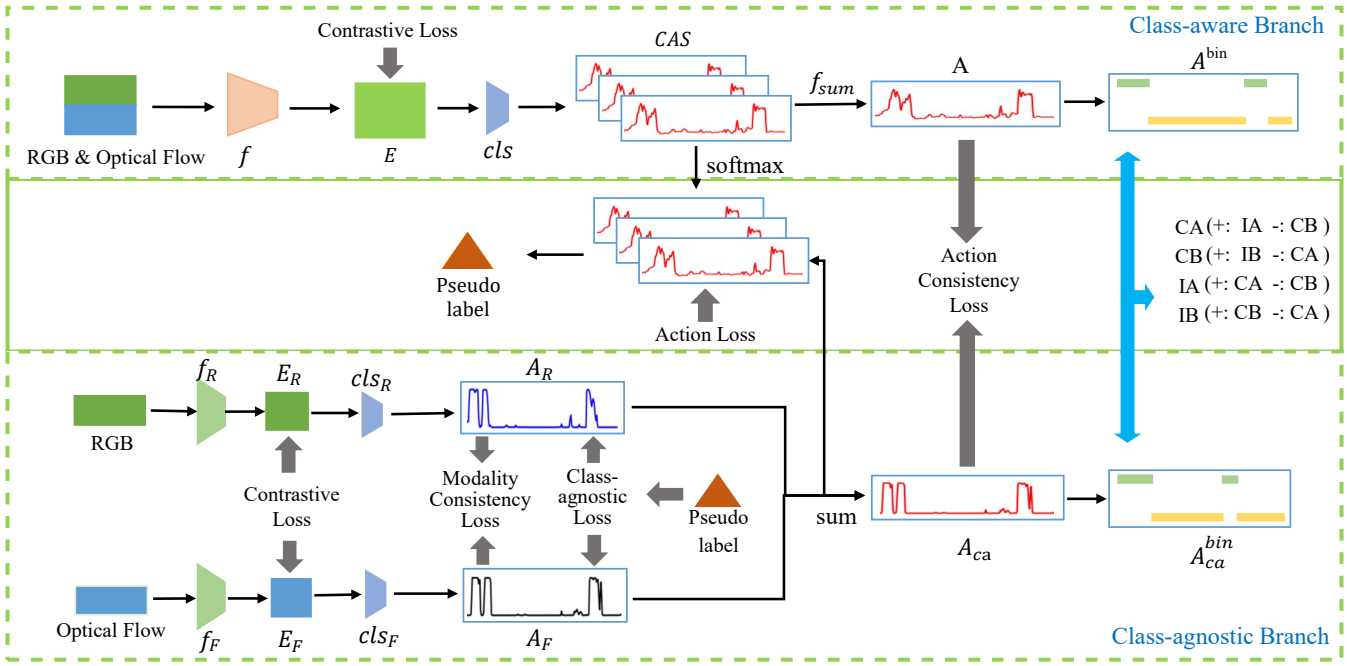


Figure 2: The framework of the proposed AICL. We first construct a two-branch baseline model, consisting of a class-aware branch and a class-agnostic branch. Then we compare the $A^{bin}$ and $A^{bin}_{ca}$ to obtain consistent action segments($CA$), consistent background segments($CB$), inconsistent action segments ($IA$), and inconsistent background segments($IB$). Finally, we employ a novel contrastive learning scheme, which explores the relationships between the above segments to improve their representations.

**Multimodal Consistency Constraints.** The motivation for the multimodal consistency loss is that there is a large noise in the hard pseudo-label. In order to alleviate the impact of the label noise on network training, we use late fusion on the class-agnostic branch to predict the results of the two modalities. The predicted scores of the two modalities can serve as soft pseudo-labels for each other. The multimodal consistency loss is added to promote collaborative training of different modalities which can help to complement each other in a proper way. Here we use the mean squared error to calculate the multimodal consistency loss:

$$\mathcal{L}_{mc} = \frac{\delta(A_R, \phi(A_F)) + \delta(A_F, \phi(A_R))}{2} \tag{11}$$

where $\phi(\cdot)$ represents a function that truncates the gradient of input, $\delta(\cdot)$ means mean squared error function.

## Contrastive Learning

It is difficult to learn discriminative representations with only weak supervision. Recently, contrastive learning (Hadsell, Chopra, and LeCun 2006; Chen et al. 2020; He et al. 2020; Chen and He 2021) is proven to be effective in representation learning without ground-truth labels. Motivated by this, we propose to enhance representation learning of the segments through contrastive learning, especially for ambiguous segments. Specifically, after getting $A$ and $A_{ca}$, we calculate the median of $A$ and $A_{ca}$ respectively. After that, we label the segments above the median as action and those below the median as background. That is, we divide half of the segments in the video into action and the other half into the background. We denote the division results on the two branches as $A^{bin}$ and $A^{bin}_{ca}$, respectively. Thereafter, we regard the segments that are labeled as action in both $A^{bin}$ and $A^{bin}_{ca}$ as consistent action segments ($CA$) and the segments that are labeled as background in both $A^{bin}$ and $A^{bin}_{ca}$ as consistent background segments ($CB$). When a segment is classified differently in $A^{bin}$ and $A^{bin}_{ca}$, we consider the segment to be an inconsistent segment. Among the inconsistent segments, we select the $k$ segment with the highest action scores as the inconsistency action segment($IA$), and we select the $k$ actions with the lowest action scores as the inconsistency background segment($IB$).

In this paper, as shown in Figure 2, we define four types of positive and negative sample pairs for contrastive learning: 1) positive samples of $IA$ are $CA$, and negative samples are $CB$; 2) positive samples of $IB$ are $CB$, and negative samples are $CA$; 3) positive samples of $CA$ are $IA$, and negative samples are $CB$; 4) positive samples of $CB$ are $IB$, and negative samples are $CA$. In this way, the relationships between consistent segments and inconsistent segments are comprehensively explored.

**Contrastive Loss.** Formally, we denote the query as $x$, the positive sample as $x^+$, and the negative sample as $x^-$. If $x \in E^{IA}$ then $x^+ \in E^{CA}$, $x^- \in E^{CB}$; if $x \in E^{IB}$ then $x^+ \in E^{CB}$, $x^- \in E^{CA}$; if $x \in E^{CA}$ then $x^+ \in E^{IA}$, $x^- \in E^{CB}$; if $x \in E^{CB}$ then $x^+ \in E^{IB}$, $x^- \in E^{CA}$. In

the class-agnostic branch, $E^{XX}$ means $E_R^{XX}$ or $E_F^{XX}$. $XX$ stands for $CA/CB/IA/IB$.

We compute the contrastive loss using the method in MoCo (He et al. 2020):

$$\mathcal{L}(E) = \sum_{x \in E} -\log \frac{exp\left(x^T \cdot x^+/\tau\right)}{\sum_{i=1}^{M} exp\left(x^T \cdot x_i^-/\tau\right) + exp\left(x^T \cdot x^+/\tau\right)} \tag{12}$$

where $\tau$ is a temperature hyper-parameter per (Wu et al. 2018), $x^T$ represents the transpose of $x$ and $M$ is the number of negative samples.

The contrastive loss used in our method is as follows:

$$\mathcal{L}_{con} = \mathcal{L}\left(E^{IA}\right) + \mathcal{L}\left(E^{IB}\right) + \mathcal{L}\left(E^{CB}\right) + \mathcal{L}\left(E^{CA}\right) \tag{13}$$

Like $\mathcal{L}_{con}$, we can compute the contrastive loss on the RGB stream and optical stream of the class-agnostic branch, i.e., $\mathcal{L}_{con_R}$ and $\mathcal{L}_{con_F}$. The final contrastive loss $\mathcal{L}_c$ is obtained by aggregating $\mathcal{L}_{con}$, $\mathcal{L}_{con_R}$, $\mathcal{L}_{con_F}$:

$$\mathcal{L}_c = \mathcal{L}_{con} + \mathcal{L}_{con_R} + \mathcal{L}_{con_F} \tag{14}$$

**Action Consistency Constraints.** Since the class-aware branch and the class-agnostic branch are trained in different manners, as the training progresses, the difference between the prediction results of the two branches will gradually increase, as shown in Figure 3. If the predictions of the two branches are too different, we will not be able to select enough consistent action/background samples to sustain contrastive learning. In order to maintain contrastive learning, we propose the two-branch action consistency constraint. Specifically, we compute the action consistency loss with the mean squared error, as follows:

$$\mathcal{L}_{ac} = \frac{\delta(A_{ca}, \phi(A)) + \delta(A, \phi(A_{ca}))}{2} \tag{15}$$

where $\phi(\cdot)$ represents a function that truncates the gradient of input, $\delta(\cdot)$ means mean squared error.

## Discussion

**Comparison with other two-branch models.** The complementarity between the class-aware and the class-agnostic branches is exploited in many works(Lee, Uh, and Byun 2020; Huang, Wang, and Li 2021; Ma et al. 2021; Qu et al. 2021). They mainly focus on the fusion of the two branches. For example, (Qu et al. 2021) proposes to combine the results of both branches to improve the performance in the test phase. In contrast, we aim to make use of the difference between the two branches to select segments for contrastive learning rather than fusing these two branches.

**Comparison with other methods with contrastive learning.** Previous work, such as CoLA (Zhang et al. 2021) also introduces contrastive learning for WTAL. It first delineates the initial action boundary by using the coarse action score predicted by the network, then proposes a dilation and erosion operation, which selects the inner segments near the action boundary as the hard action segments and the outer segments as the hard background segments. However, due to the low quality of the action boundary and the artificially set fixed degree for dilation and erosion, it easily leads to inaccurate action/background decisions for hard segments, thereby reducing the effect of contrastive learning. In this work, we construct two branches with different structures and different functions to dynamically find the ambiguous segments instead of just relying on the rough action score. FTCL takes similar proposals as positive samples and dissimilar proposals as negative samples. DCC divides the video into multiple regions and generates denoised pseudo labels for each region. According to the pseudo labels, positive samples of the same category and negative samples of different categories are determined. In a word, our work is based on segment-level contrastive learning, FTCL is based on proposal-level, and DCC is based on region-level.

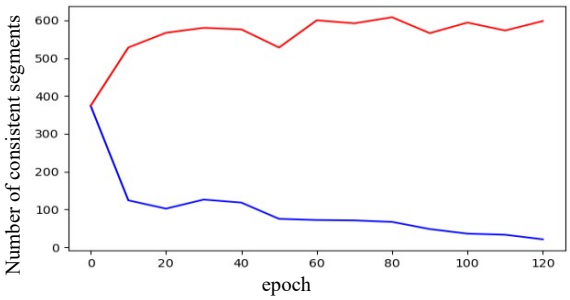


Figure 3: The change process of the number of consistent segments during training on THUMOS14. The blue curve represents training without action consistency constraints, and the red curve represents training with action consistency constraints.

## Training and Inference

**Final Objective Function.** To train the whole framework end-to-end, we aggregate all aforementioned objective functions to form the final objective function:

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}_{ca} + \gamma_1 \mathcal{L}_c + \gamma_2 \mathcal{L}_{mc} + \gamma_3 \mathcal{L}_{ac} \tag{16}$$

where the $\gamma_1$, $\gamma_2$ and $\gamma_3$ are hyperparameters.

**Inference.** In the test stage, following (Ma et al. 2021), we use $A_{score}$ to localize actions. Firstly, we calculate the video-level categorical probabilities in the class-aware branch, which indicate the possibility of each action class happening in the given video. Then we aggregate action possibilities of the class-agnostic branch and get $A_{score}$. We apply multiple thresholds $0 < \theta < 1$. All instances where selection probability is above the threshold $A_{score} > \theta$ are considered selected, and we take all consecutive sequences as proposal candidates. Finally, Non-Maximum Suppression (NMS) is applied to remove duplicated proposals.

# Experiments

## Datasets and Metrics

**Datasets.** There are 200 validation videos and 213 test videos of 20 action classes in the THUMOS14(Idrees et al.

| Supervision | Method | mAP@IoU (%) 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | AVG 0.1:0.7 |
|---|---|---|---|---|---|---|---|---|---|
| Fully | S-CNN (Shou, Wang, and Chang 2016) | 47.7 | 43.5 | 36.3 | 28.7 | 19.0 | 10.3 | 5.3 | 24.3 |
| | TAL-Net (Chao et al. 2018) | 59.8 | 57.1 | 53.2 | 48.5 | 42.8 | 33.8 | 20.8 | 45.1 |
| | P-GCN(Zeng et al. 2019) | 69.5 | 67.5 | 63.6 | 57.8 | 49.1 | - | - | - |
| | TRA(Zhao et al. 2022) | **73.7** | **72.6** | **70.0** | **64.3** | **57.4** | **46.2** | **31.1** | **59.3** |
| Weakly | DML (Islam and Radke 2020) | 62.3 | - | 46.8 | - | 29.6 | - | 9.7 | - |
| | A2CL-PT (Min and Corso 2020) | 61.2 | 56.1 | 48.1 | 39.0 | 30.1 | 19.2 | 10.6 | 37.8 |
| | HAM-Net (Islam, Long, and Radke 2021) | 65.9 | 59.6 | 52.2 | 43.1 | 32.6 | 21.9 | 12.5 | 39.8 |
| | UM (Lee et al. 2021) | 67.5 | 61.2 | 52.3 | 43.4 | 33.7 | 22.9 | 12.1 | 41.9 |
| | CoLA (Zhang et al. 2021) | 66.2 | 59.5 | 51.5 | 41.9 | 32.2 | 22.0 | 13.1 | 40.9 |
| | ASL (Ma et al. 2021) | 67.0 | - | 51.8 | - | 31.1 | - | 11.4 | - |
| | $CO_2$-Net (Hong et al. 2021) | 70.1 | 63.6 | 54.5 | 45.7 | **38.3** | **26.4** | 13.4 | 44.6 |
| | DCC (Li et al. 2022) | 69.0 | 63.8 | 55.9 | 45.9 | 35.7 | 24.3 | 13.7 | 44.0 |
| | ACGNet (Yang, Qin, and Huang 2022) | 68.1 | 62.6 | 53.1 | 44.6 | 34.7 | 22.6 | 12.0 | 42.5 |
| | FTCL(Gao, Chen, and Xu 2022) | 69.6 | 63.4 | 55.2 | 45.2 | 35.6 | 23.7 | 12.2 | 43.6 |
| | RSKP (Huang, Wang, and Li 2022) | 71.3 | 65.3 | 55.8 | 47.5 | 38.2 | 25.4 | 12.5 | 45.1 |
| | **AICL(ours)** | **73.1** | **67.8** | **58.2** | **48.7** | 36.9 | 25.3 | **14.9** | **46.4** |

Table 1: Comparisons of AICL with other methods on the THUMOS14 dataset.

| Sup. | Method | mAP@IoU (%) 0.5 | 0.75 | 0.95 | AVG |
|---|---|---|---|---|---|
| Fully | SSN(2017) | 41.3 | 27.0 | 6.1 | 26.6 |
| Weakly | BaSNet (2020) | 38.5 | 24.2 | 5.6 | 24.3 |
| | DGAM (2020) | 41.0 | 23.5 | 5.3 | 24.4 |
| | Gong (2020) | 40.0 | 25.0 | 4.6 | 24.6 |
| | TSCN (2020) | 37.6 | 23.7 | 5.7 | 23.6 |
| | HAM-Net (2021) | 41.0 | 24.8 | 5.3 | 25.1 |
| | UM (2021) | 41.2 | 25.6 | 6.0 | 25.9 |
| | ACSNet (2021b) | 40.1 | 26.1 | **6.8** | 26.0 |
| | $CO_2$-Net (2021) | 43.3 | 26.3 | 5.2 | 26.4 |
| | ACGNet (2022) | 41.8 | 26.0 | 5.9 | 26.1 |
| | **AICL(ours)** | **49.6** | **29.1** | 5.9 | **29.9** |

Table 2: Comparison on the ActivityNet1.2. AVG means average mAP from IoU 0.5 to 0.95 with 0.05 increment.

| Sup. | Method | mAP@IoU (%) 0.5 | 0.75 | 0.95 | AVG |
|---|---|---|---|---|---|
| Fully | BSN(2018) | 46.5 | 30.0 | 8.9 | 30.0 |
| | GTAN (2019) | 52.6 | 34.1 | 8.9 | 34.3 |
| Weakly | TS-PCA (2021a) | 37.4 | 23.5 | 5.9 | 23.7 |
| | UGCT (2021) | 39.1 | 22.4 | 5.8 | 23.8 |
| | AUMN (2021) | 38.3 | 23.5 | 5.2 | 23.5 |
| | FAC-Net (2021) | 37.6 | 24.2 | 6.0 | 24.0 |
| | DCC (2022) | 38.8 | 24.2 | 5.4 | 24.3 |
| | FTCL (2022) | 40.0 | 24.3 | **6.4** | 24.8 |
| | RSKP (2022) | 40.6 | 24.6 | 5.9 | 25.0 |
| | **AICL(ours)** | **44.2** | **27.4** | 5.8 | **27.6** |

Table 3: Comparison on the ActivityNet1.3. AVG means average mAP from IoU 0.5 to 0.95 with 0.05 increment.

2017). Following (Ma et al. 2021; Lee, Uh, and Byun 2020; Liu et al. 2021b), we use 200 validation videos to train our framework and 213 test videos for testing. ActivityNet1.2(Caba Heilbron et al. 2015) is composed of 4,819 training videos, 2,383 validation videos, and 2,489 test videos of 100 action classes. ActivityNet1.3(Caba Heilbron et al. 2015) contains 10,024 training videos and 4,926 validation videos from 200 action categories. Following previous work (Gao, Chen, and Xu 2022; Li et al. 2022; Huang, Wang, and Li 2022), we train on the training set and test on the validation set.

**Evaluation Metrics.** We follow the standard evaluation protocol by reporting mean Average Precision (mAP) values under different intersections over union (IoU) thresholds.

**Implementation Details.** Following existing methods, we use I3D (Carreira and Zisserman 2017) model pretrained on Kinetics (Carreira and Zisserman 2017) dataset as the RGB and optical flow feature extractors. In the training and testing stage, we randomly sample 750 snippets for the THUMOS14 dataset. In the structure diagram of AICL, the $f$, $f_R$, and $f_F$ are constructed with convolution layer and RELU activations. The output dimensions of the convolution layer are 512 for THUMOS14. The $cls$ is constructed with a convolution layer whose output dimension is 20, while the kernel size is 1. The $cls_R$ and $cls_F$ are constructed with a convolution layer whose output dimensions are 1, while the kernel size is 1. We set noise tolerance $q$ = 0.7 for both datasets, and use instance selection parameters $k = T/8$ for THUMOS14. We set $\gamma$ to be 1/3 for the $A_{score}$ in THUMOS14. We set $K = T/20$ to choose the number of inconsistent action and background segments. We set $\gamma_1$ = 0.01 for THUMOS14.

| Exp | $\mathcal{L}_a$ | $\mathcal{L}_{ca}$ | $\mathcal{L}_c$ | $\mathcal{L}_{mc}$ | $\mathcal{L}_{ac}$ | AVG (%) |
|---|---|---|---|---|---|---|
| 1 | ✓ | | | | | 31.9 |
| 2 | ✓ | ✓ | | | | 43.2 |
| 3 | ✓ | ✓ | ✓ | | | 44.8 |
| 4 | ✓ | ✓ | | ✓ | | 44.6 |
| 5 | ✓ | ✓ | ✓ | ✓ | | 46.0 |
| 6 | ✓ | ✓ | ✓ | ✓ | ✓ | 46.4 |

Table 4: Ablation studies of our algorithm.

| Method | Accuracy | |
|---|---|---|
| | $A^{bin}$ | $A^{bin}_{ca}$ |
| baseline($\mathcal{L}_a$ + $\mathcal{L}_{ca}$) | 37.3% | 65.7% |
| AICL | 60.3% | 65.3% |

Table 5: The accuracy of the action-background binary classification of $A^{bin}$ and $A^{bin}_{ca}$ on THUMOS14.

## Comparison with State-of-the-art Methods

We first compare ours with current state-of-the-art W-TAL methods and several fully supervised methods. We report the results in Table 1, Table 2 and Table 3.

**Results on THUMOS14.** Table 1 shows the performance of different methods on THUMOS14. Our method outperforms all the methods in most IoU metrics. Especially in the low IoU part, it is about 2% higher than other methods.

**Results on ActivityNet1.2 & ActivityNet1.3.** Table 2 and Table 3 show the localization performance of different methods on ActivityNet1.2 and ActivityNet1.3. We can find that our method significantly outperforms other methods on ActivityNet1.2 and ActivityNet1.3.

## Ablation Study

**The Effect of Each Object Functions.** Each component in the final objective Eq. 16 plays an important role in our framework. To verify the effectiveness of each objective function, we conduct related ablation studies and report results in Table 4. Note that, we set "Exp2" as our baseline that uses multiple-instance learning action loss($\mathcal{L}_a$) and class-agnostic loss($\mathcal{L}_{ca}$). From the results, we can see that all the proposed loss terms are effective.

**The Accuracy of $A^{bin}$ & $A^{bin}_{ca}$.** In order to further demonstrate the effectiveness of our method, we compare both $A^{bin}$ and $A^{bin}_{ca}$ on the baseline and AICL with ground-truth frame-level labels. It can be clearly seen from Table 5 that AICL leads to a significant improvement in the accuracy of $A^{bin}$ compared to the baseline.

**The Effect of Fusion Method.** Table 6 shows the action localization performance of the two branches using different fusion methods for RGB features and optical flow features on the THUMOS14. We can observe that 1) the late fusion is more suitable for the class-agnostic branch. 2) It is better to use different fusion strategies for the different branches.

| Branch | Fusion method | | | |
|---|---|---|---|---|
| class-aware | early | late | late | early |
| class-agnostic | early | early | late | late |
| AVG(%) | 40.7 | 41.6 | 45.7 | 46.4 |

Table 6: Comparing feature fusion methods for RGB and optical flow with different branches.

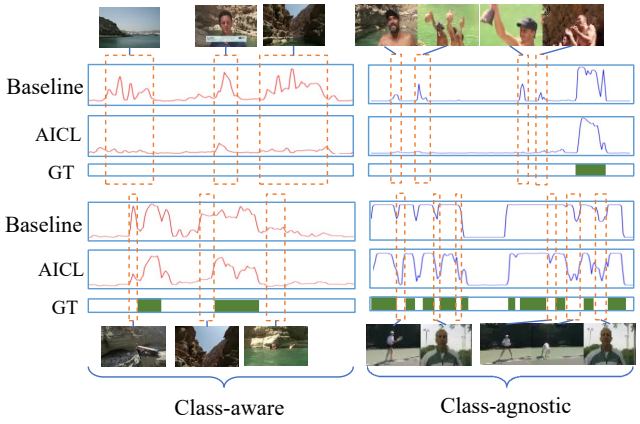


Figure 4: Qualitative results of two examples of "CliffDiving" and "TennisSwing" on THUMOS14. We set "Exp2" in Table 4 as the baseline.

## Visual Results

To better illustrate our method, Figure 4 visualizes some qualitative results. We compare the class-aware and class-agnostic branches with the baseline (i.e. "Exp2" in Table 4) respectively. In the class-aware branch, our AICL has a significant inhibitory effect on incorrect scene-related action activation compared to the baseline. In the class-agnostic branch, AICL effectively suppresses label-category-independent action activation compared to baseline. By showing the two branches separately, it is obvious that scene-related errors and action-related errors are effectively mitigated by AICL.

## Conclusion

In this work, we propose a novel WTAL approach named AICL. We first define consistent/inconsistent action/background segments based on the comparison between the class-aware branch and the class-agnostic branch. Then we leverage the consistent segments to boost the representations of the inconsistent segments with the aid of contrastive learning. In addition, we also design action consistency constraints and multimodal consistency constraints to facilitate mutual learning of different branches. Finally, we conduct extensive experiments on three challenging benchmarks to verify the effectiveness of our proposed AICL.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62176246 and Grant 61836008. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

## References

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 961–970. IEEE.

Carbonneau, M.-A.; Cheplygina, V.; Granger, E.; and Gagnon, G. 2018. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 329–353.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 6299–6308.

Chao, Y.-W.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R. 2018. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*, 15750–15758.

Gao, J.; Chen, M.; and Xu, C. 2022. Fine-grained Temporal Contrastive Learning for Weakly-supervised Temporal Action Localization. In *CVPR*, 19999–20009.

Gong, G.; Wang, X.; Mu, Y.; and Tian, Q. 2020. Learning Temporal Co-Attention Models for Unsupervised Video Action Localization. In *CVPR*.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 297–304. JMLR Workshop and Conference Proceedings.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 1735–1742. IEEE.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

Hong, F.-T.; Feng, J.-C.; Xu, D.; Shan, Y.; and Zheng, W.-S. 2021. Cross-modal consensus network for weakly supervised temporal action localization. In *Proceedings of the 29th ACM International Conference on Multimedia*, 1591–1599.

Huang, L.; Wang, L.; and Li, H. 2021. Foreground-Action Consistency Network for Weakly Supervised Temporal Action Localization. In *ICCV*, 8002–8011. IEEE.

Huang, L.; Wang, L.; and Li, H. 2022. Weakly Supervised Temporal Action Localization via Representative Snippet Knowledge Propagation. In *CVPR*, 3272–3281.

Idrees, H.; Zamir, A. R.; Jiang, Y.-G.; Gorban, A.; Laptev, I.; Sukthankar, R.; and Shah, M. 2017. The THUMOS challenge on action recognition for videos "in the wild". *CVIU*, 1–23.

Islam, A.; Long, C.; and Radke, R. 2021. A hybrid attention mechanism for weakly-supervised temporal action localization. In *AAAI*, 2, 1637–1645.

Islam, A.; and Radke, R. 2020. Weakly supervised temporal action localization using deep metric learning. In *CVPR*, 547–556.

Lee, P.; Uh, Y.; and Byun, H. 2020. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, 07, 11320–11327.

Lee, P.; Wang, J.; Lu, Y.; and Byun, H. 2021. Weakly-supervised temporal action localization by uncertainty modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3, 1854–1862.

Li, J.; Yang, T.; Ji, W.; Wang, J.; and Cheng, L. 2022. Exploring Denoised Cross-Video Contrast for Weakly-Supervised Temporal Action Localization. In *CVPR*, 19914–19924.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 3–19. Springer.

Liu, Y.; Chen, J.; Chen, Z.; Deng, B.; Huang, J.; and Zhang, H. 2021a. The Blessings of Unlabeled Background in Untrimmed Videos. In *CVPR*, 6176–6185. IEEE.

Liu, Z.; Wang, L.; Zhang, Q.; Tang, W.; Yuan, J.; Zheng, N.; and Hua, G. 2021b. Acsnet: Action-context separation network for weakly supervised temporal action localization. In *AAAI*, 3, 2233–2241.

Long, F.; Yao, T.; Qiu, Z.; Tian, X.; Luo, J.; and Mei, T. 2019. Gaussian temporal awareness networks for action localization. In *CVPR*.

Luo, W.; Zhang, T.; Yang, W.; Liu, J.; Mei, T.; Wu, F.; and Zhang, Y. 2021. Action Unit Memory Network for Weakly Supervised Temporal Action Localization. In *CVPR*, 9969–9979. IEEE.

Ma, J.; Gorti, S. K.; Volkovs, M.; and Yu, G. 2021. Weakly supervised action selection learning in video. In *CVPR*, 7587–7596.

Min, K.; and Corso, J. J. 2020. Adversarial background-aware loss for weakly-supervised temporal activity localization. In *ECCV*.

Nawhal, M.; and Mori, G. 2021. Activity graph transformer for temporal action localization. *arXiv preprint arXiv:2101.08540*.

Nguyen, P.; Liu, T.; Prasad, G.; and Han, B. 2018. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, 6752–6761.

Nguyen, P. X.; Ramanan, D.; and Fowlkes, C. C. 2019. Weakly-supervised action localization with background modeling. In *ICCV*, 5502–5511.

Qu, S.; Chen, G.; Li, Z.; Zhang, L.; Lu, F.; and Knoll, A. 2021. Acm-net: Action context modeling network for weakly-supervised temporal action localization. *arXiv preprint arXiv:2104.02967*.

Shi, B.; Dai, Q.; Mu, Y.; and Wang, J. 2020. Weakly-Supervised Action Localization by Generative Attention Modeling. In *CVPR*.

Shou, Z.; Wang, D.; and Chang, S.-F. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Yang, W.; Zhang, T.; Yu, X.; Qi, T.; Zhang, Y.; and Wu, F. 2021. Uncertainty guided collaborative training for weakly supervised temporal action detection. In *CVPR*, 53–63.

Yang, Z.; Qin, J.; and Huang, D. 2022. ACGNet: Action Complement Graph Network for Weakly-Supervised Temporal Action Localization. In *AAAI*, 3, 3090–3098.

Yu, T.; Ren, Z.; Li, Y.; Yan, E.; Xu, N.; and Yuan, J. 2019. Temporal structure mining for weakly supervised action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5522–5531.

Zeng, R.; Huang, W.; Tan, M.; Rong, Y.; Zhao, P.; Huang, J.; and Gan, C. 2019. Graph convolutional networks for temporal action localization. In *ICCV*.

Zhai, Y.; Wang, L.; Tang, W.; Zhang, Q.; Yuan, J.; and Hua, G. 2020. Two-stream consensus network for weakly-supervised temporal action localization. In *ECCV*, 37–54. Springer.

Zhang, C.; Cao, M.; Yang, D.; Chen, J.; and Zou, Y. 2021. Cola: Weakly-supervised temporal action localization with snippet contrastive learning. In *CVPR*, 16010–16019.

Zhang, Z.; and Sabuncu, M. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In *ICCV*, 2914–2923.

Zhao, Y.; Zhang, H.; Gao, Z.; Guan, W.; Nie, J.; Liu, A.; Wang, M.; and Chen, S. 2022. A temporal-aware relation and attention network for temporal action localization. *IEEE Transactions on Image Processing*, 31: 4746–4760.