Self-Supervised Video Action Localization with Adversarial Temporal Transforms

Guoqiang Gong[1] , **Liangfeng Zheng**[1] , **Wenhao Jiang**[2] , **Yadong Mu**[1*]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Tencent AI Lab
{gonggq, zhengliangfeng, myd}@pku.edu.cn, cswhjiang@gmail.com

Abstract

Weakly-supervised temporal action localization aims to locate intervals of action instances with only video-level action labels for training. However, the localization results generated from video classification networks are often not accurate due to the lack of temporal boundary annotation of actions. Our motivating insight is that the temporal boundary of action should be stably predicted under various temporal transforms. This inspires a self-supervised equivariant transform consistency constraint. We design a set of temporal transform operations, including naive temporal down-sampling to learnable attention-piloted time warping. In our model, a localization network aims to perform well under all transforms, and another policy network is designed to choose a temporal transform at each iteration that adversarially brings localization result inconsistent with the localization network's. Additionally, we devise a self-refine module to enhance the completeness of action intervals harnessing temporal and semantic contexts. Experimental results on THUMOS14 and ActivityNet demonstrate that our model consistently outperforms the state-of-the-art weakly-supervised temporal action localization methods.

1 Introduction

Temporal action localization (TAL) in untrimmed videos is one of the most challenging tasks in video understanding. In real-world applications, untrimmed videos usually contain multiple action instances and irrelevant background scenes. The aim of temporal action localization is to locate the temporal boundary and predict the action category for each action instance in a video. To learn an effective action localization model, most existing temporal action localization methods utilize fine-grained supervision, which requires manually annotated temporal boundaries and action category label for each action instance. However, labeling the temporal boundary of an action instance is time-consuming. In this paper, we focus on weakly-supervised temporal action localization, where only video-level action category labels are available.

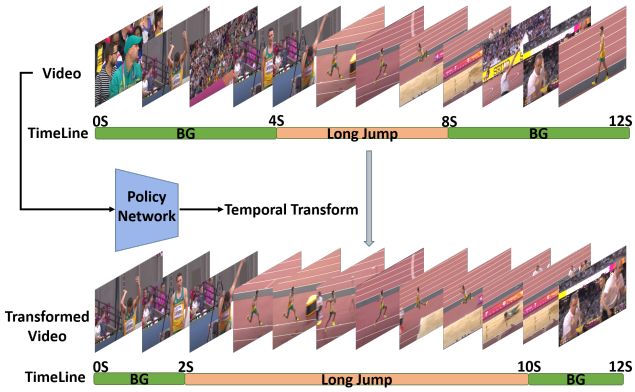

Figure 1: Illustration of our motivation. 'BG' means background. We learn to generate adversarial temporal transforms. A good model should ensure video-level classification to be invariant, and the temporal boundary of an action equivariant under all temporal transforms.

To tackle the weakly-supervised TAL task, most existing works [Nguyen *et al.*, 2018; Narayan *et al.*, 2019; Yu *et al.*, 2019] fall into a multiple-instance-learning framework. In specific, a video is treated as a bag of frames and fed into video-level classification networks. By enforcing the accuracy of video-level predictions, a weakly-supervised TAL model learns to generate a class activation sequence (CAS) for each video, which essentially indicates how likely each frame belongs to an action class [Shou *et al.*, 2018]. Action instances can then be temporally localized based on the CAS. However, without supervision from temporal boundaries of actions, CAS is empirically observed to be often incomplete (tends to cover only the most discriminative part of the action) and noisy (often incorrectly activates background moments).

Figure 1 illustrates the motivation of this work. The performance of a weakly-supervised TAL model heavily relies on the quality of CAS. We aim to address a rarely-explored aspect in weakly-supervised TAL: self-supervision brought by temporal transforms on CAS. Given an untrimmed video that contains several actions, the resultant CAS can drastically change after applying some temporal transforms to a

*Corresponding author.

video (such as fast forwarding selected segment of the video while keeping other segments unchanged). It does not bring a severe issue for video-level prediction, since it is invariant to these transforms. However, the stability of CAS is crucial for precisely de-limiting the action temporal boundary. To obtain good performance, we would expect the CAS to be strongly equivariant with respect to numerous temporal transforms.

To mitigate the supervision gap between weakly and fully supervised temporal action localization, we propose a self-supervised equivariant transform consistency constraint that minimizes the discrepancy between the CAS of the original / temporally-transformed videos. In particular, we design an action localization network with a Siamese architecture, comprised of two sub-networks with the identical design that read the feature sequences of the original / temporally-transformed videos, respectively. Without loss of generality, this work only investigates three types of temporal transforms including resize, window warp, and attention-based time warp. A policy network is learned to select a temporal transform at each iteration. Critically, the localization and policy networks shall operate adversarially to each other, similar to Generative Adversarial Networks (GAN). The localization network desires the two Siamese sub-networks return transform-equivariant class activation sequences, while the policy network selects a transform that potentially maximizes the inconsistency between original / temporally-transformed videos. In addition, to enhance the completeness of the TAL results, we also propose a self-refine module that utilizes both temporal and semantic contexts to generate more integral action instances.

The technical contributions of this work can be summarized as below: (1) To our best knowledge, this work represents the first attempt to explore self-supervised equivariant consistency regularization for weakly-supervised temporal action localization. To generate temporal transformed videos, we propose attention based time warp methods and design a policy network to select transform operations for each video; (2) We propose a self-refine module to explicitly leverage the temporal and semantic context information to obtain a more complete temporal interval of actions; (3) Comprehensive evaluations are conducted on two challenging video benchmarks: THUMOS14 and ActivityNet. Our method re-calibrates the state-of-the-art performance on both benchmarks by large margins.

2 Related Work

Weakly-supervised action localization. It learns to localize activities inside videos with only video-level action category labels available. Most of the relevant approaches adopt the multiple instance learning framework. UntrimmedNet [Wang *et al.*, 2017a] proposes a selection module to rank clip proposals and locate action instances. CMCS [Liu *et al.*, 2019] improves UntrimmedNet by using a multi-branch network with diversity loss to model action completeness. 3C-Net [Narayan *et al.*, 2019] implies that multi-label center loss and action counting loss can be used to decrease intra-class variations and increase the separability of adjacent action instances. BM [Nguyen *et al.*, 2019] proposes a background-aware loss to explicitly model background content. DGAM [Shi *et al.*, 2020] proposes a generative attention mechanism to separate action and context frames. TSCN [Zhai *et al.*, 2020] generates frame-level pseudo labels by combining the predictions of RGB and flow streams. Although these methods generally achieve promising results, they only process the video at a fixed temporal resolution, ignoring the considerable variation of action duration. More importantly, previous works have rarely studied the equivariant property between the temporal boundary of an action and temporal transforms.

Self supervised learning (SSL). It has received increasing attention, noting the potential of learning effective semantic feature representations without human annotations. Prior study has explored different kinds of self-supervision, such as prediction of the spatial context [Doersch *et al.*, 2015], colorization [Zhang *et al.*, 2016], and transition equivalence [Gidaris *et al.*, 2018; Noroozi *et al.*, 2017]. Recent SSL works on videos mainly focus on spatio-temporal orders of videos. For example, the frame order in videos is the main clue utilized in [Misra *et al.*, 2016]. Other types of temporal context include its conjunction of spatial context [Wang *et al.*, 2017b], as well as the use of statistics on spatio-temporal co-occurrence [Isola *et al.*, 2015]. In all the works, [Wang *et al.*, 2020] is most close to our work. It uses an attention mechanism to do self-supervised learning on semantic segmentation. However, the self-supervision used in [Wang *et al.*, 2020] focuses on simple scale transforms in the spatial domain. Differently, our method operates temporally with learnable video transforms. We formulate it in an adversarial fashion by designing a policy network to select the most challenging temporal transform at each iteration. To our best knowledge, we are the first to explore SSL in the weakly-supervised TAL task.

3 Our Method

During training, we are provided with a training set of untrimmed videos $V = \{v_i, y_i\}_{i=1}^{M}$, where M is the number of videos, $y_i \in \{0,1\}^C$ denotes the video-level category label of video v_i, C is the number of action categories. At test time, the output for each test video is a set of localized action instances (b_j, e_j, c_j, q_j), where b_j and e_j denote the start and end time, c_j refers to the predicted action category, and q_j is the confidence score.

3.1 Video Feature Extraction

To extract feature sequence, we first divide an untrimmed video into a set of snippets, each of which contains several consecutive frames. As in previous works [Narayan *et al.*, 2019; Nguyen *et al.*, 2019; Shi *et al.*, 2020], the RGB and flow I3D [Carreira and Zisserman, 2017] models pretrained on Kinetics [Carreira and Zisserman, 2017] are utilized to extract two-stream video features for each snippet. Let $X \in \mathbb{R}^{T \times D}$ denote the RGB or flow feature sequence, where T denotes the count of snippets and D is the dimension of features.

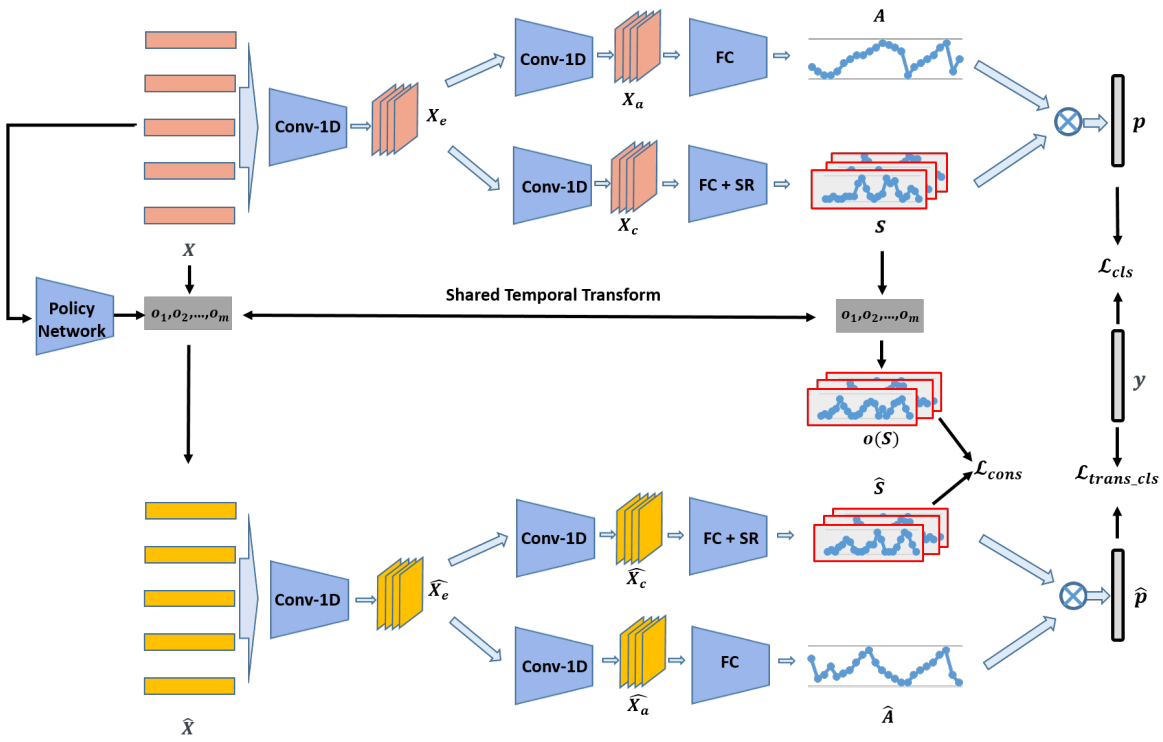

Figure 2: Architecture of our proposed model. 'FC' means fully-connected layer,'SR' means Self-refine module. See text for more details.

3.2 Architecture Overview

As shown in Figure 2, we propose an action localization network with a Siamese design, containing two sub-networks Net_1 and Net_2 that share the same structure and parameters. The inputs of Net_1 and Net_2 are video feature sequences $X, \hat{X} = o(X)$ extracted from a video and its temporally-transformed version, where o denotes a temporal transform operation. A policy network is included to choose one from a set of pre-defined temporal transform operations O.

For the Siamese net, let us take Net_1 to describe the computational flow. The initial video feature X first goes through some additional temporal convolutions to better adapt to the current task, obtaining the embedding feature X_e. After that, X_e is fed to a classification branch and an attention branch, rendering features X_a and X_c respectively. Each branch is composed of a temporal convolution layer and a fully connected layer as the net head. The attention branch generates temporal attention weights $A \in \mathbb{R}^{T\times 1}$. For the classification branch, its coarse output is sent to a self-refine (SR) module, getting the CAS $S \in \mathbb{R}^{T\times C}$. Eventually, A and S are joined to predict the class probability $p = softmax(A^\intercal S)$ in the video level. Similar treatment for Net_2.

The learning of action localization network is guided by:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{trans\_cls} + \alpha\mathcal{L}_{cons}, \tag{1}$$

where $\mathcal{L}_{cls}$, $\mathcal{L}_{trans\_cls}$ are the classification losses of Net_1 and Net_2 respectively. $\mathcal{L}_{cons}$ is self-supervised transform consistency loss. α is a weighting coefficient. We will elaborate on the details of each term in following sections.

3.3 Self-refine Module

The classification branch generates the CAS for each snippet. However, the initial CAS tends to be incomplete or noisy. To remedy it, we propose a self-refine module that utilizes both temporal and semantic context information.

For a video v that contains T snippets, we first construct a fully connected graph G. Nodes in G are the T snippets of v. Let $E \in \mathbb{R}^{T\times T}$ be the adjacency matrix of G. $E_{i,j}$ is the edge weight between node i and j. Intuitively, temporally neighboring snippets are more likely to have correlated content. We define the temporal weight $E^{tem}_{i,j}$ of edge i, j as:

$$E^{tem}_{i,j} = e^{-\frac{|i-j|^2}{2\sigma^2}}, \tag{2}$$

where σ is empirically set as $T/10$ in all experiments. For the semantic similarity of snippets i and j, we evaluate it by measuring the Cosine value:

$$E^{sem}_{i,j} = ReLU\left(\frac{X_r[i]^\intercal X_r[j]}{\|X_r[i]\| \cdot \|X_r[j]\|}\right), \tag{3}$$

where $X_r = Conv1D(X_c) \in \mathbb{R}^{T\times D_1}$ is an enhanced feature with learnable 1-D convolution.

The final inter-node similarity $E_{i,j}$ is calculated by multiplication:

$$E_{i,j} = E^{tem}_{i,j} * E^{sem}_{i,j}. \tag{4}$$

For numerical tractability, weights of the outbound edges of each node are normalized to be 1. After constructing the adjacency matrix E, a random walk routine is called to refine

the coarse CAS S of the classification branch. In implementation, S is updated iteratively via $S \longleftarrow E \times S$. We empirically find that an overall iteration of 3 times will strike a good balance of accuracy and computational complexity.

3.4 Self-supervised Equivariant Consistency

Human visual perception shows good consistency for temporal action localization tasks when they watch the video at different playback rates. For example, when we watch a video that contains a high jump action, the action category of the video will not change, yet the duration and temporal boundary of the high jump action will change as the playback rate varies. State differently, the action category of the video is invariant to different playback rates, while the temporal boundary of action is equivariant. To make CAS S have the equivariant property, this work proposes a self-supervised equivariant consistency loss L_{cons}.

Given a video v with action category c. Suppose the input video undergoes a temporal transform defined by $o(\cdot)$, namely $\hat{X} = o(X)$ for the input video feature X. Recall that $S, \hat{S}$ are the final class activation sequences for the original / transformed videos. Critically, if the activation sequence S were equivariant with respect to $o(\cdot)$, there shall be high consistency with $\hat{S}$. Let $S^o = o(S)$ and we only slice the profile of the ground-truth class c, namely $S^o_c = softmax(S^o[:, c])$, $\hat{S}_c = softmax(\hat{S}[:, c])$. Here the $softmax$ is used to get the action class distribution along the temporal axis. The self-supervised equivariant consistency loss $\mathcal{L}_{cons}$ is defined as:

$$\mathcal{L}_{cons} = KL(S^o_c \parallel \hat{S}_c) + KL(\hat{S}_c \parallel S^o_c), \tag{5}$$

where KL denotes the Kullback–Leibler divergence.

3.5 Adversarial Temporal Transform

To make the idea of self-supervision feasible, we first define a bank of candidate temporal transforms, which can be divided into the following categories:

1.Resize. Similar to the resize operation in image processing, the resize operation performs up-sampling or down-sampling the entire video feature sequence along the time dimension uniformly.

2.Window warp. It divides the feature sequence into some non-overlapping sub-sequences. A sub-sequence will be chosen for either temporal stretching or contraction.

3.Attention based time warp. It re-samples the video feature map following a probability distribution induced by the temporal attention A.

To perform time warp, we first get the upsampled video feature $X_{up} \in \mathbb{R}^{NT \times D}$ and temporal attention $A_{up} \in \mathbb{R}^{NT \times 1}$ by performing linear interpolation on original X, A. N is a parameter to control the upsampling rate. We devise three different time-warp operators: (1) attention-enhanced time warp. (2) attention-complemented time warp. (3) attention gradient enhanced time warp. The attention enhanced time warp samples T locations from the upsampled NT locations. The i-th position is sampled with a probability $dis_{enh}[i]$, where

$$dis_{enh} = softmax(A_{up}). \tag{6}$$

The temporal interval highlighted by A is often the action-occurring moment. dis_{enh} leads more dense sampling (similar to slow-motion playback) on action-related intervals and suppresses background segments (similar to fast-forwarding). In this way, we have more fine-grained view of action boundaries.

Likewise, the attention complemented time warp samples T locations following a probability dis_{comp}, where

$$dis_{comp} = \frac{1 - dis_{enh}}{NT - 1}. \tag{7}$$

The attention complemented time warp mainly samples temporal location that is overlooked by temporal attention A. A video usually contains multiple actions. The temporal attention usually focuses on the most discriminative part of a video, ignoring some action instances with short duration. Attention complemented time warp allows the network to mine more actions that are missed by current temporal attention.

The attention gradient enhanced time warp samples T locations from dis_{ga}, where

$$dis_{ga} = softmax(GA), GA[i] = A_{up}[i+1] - A_{up}[i]. \tag{8}$$

Temporal locations with large gradients often belong to the boundary of the action. By intensive sampling nearby the boundary, the start and end time of the actions can be determined more accurately.

Inspired by GANs, we design a policy network to select a transform operation o from the transform bank O, which maximizes training loss of the localization network. In practice, the policy network consists of one temporal convolution layer and two fully-connected layers. Let θ denote the parameters of policy network. Taking video feature sequence as input, policy network output the distribution $p_\theta(o)$ which defines a probability of operation o being selected. Formally, the goal of policy network is to maximize $J(\theta) = \sum_{o \in O} p_\theta(o) r(o)$, where $r(o) = \mathcal{L}_{trans\_cls} + \alpha \mathcal{L}_{cons}$ is a reward corresponding to transform operation o.

3.6 Optimization

Since some temporal transform operations are non-differentiable, it is intractable to train the localization / policy networks jointly. Instead, we train them by alternately executing the following steps: (1) Update the action localization network with loss $\mathcal{L}$ as in Eqn. 1. (2) Utilize the REINFORCE [Williams, 1992] algorithm to train the policy network.

3.7 Action Localization

During testing, only Net_1 is required for action localization. Given a test video feature sequence X, we first use Net_1 to obtain its CAS S and class probability distribution p. We utilize a two-stage threshold method to generate action localization results. First, we filter out action categories whose probabilities are below a threshold τ. For a remaining category c, we use a set of threshold values $[\alpha_0, ..., \alpha_r]$ to threshold on $S[:, c]$ respectively and generate localization proposals. Let (b_i, e_i, c, q_i) denote the i-th proposal, where b_i is the start

Supervision	Methods	mAP@IoU (%)						
		0.1	0.2	0.3	0.4	0.5	0.6	0.7
Full	SSN [Zhao *et al.*, 2017]	66.0	59.4	51.9	41.0	29.8	-	-
	TAL-Net [Chao *et al.*, 2018]	59.8	57.1	53.2	48.5	42.8	33.8	20.8
	G-TAD [Xu *et al.*, 2020]	-	-	54.5	47.6	40.2	30.8	23.4
Weak	UntrimmedNet [Wang *et al.*, 2017a]	44.4	37.7	28.2	21.1	13.7	-	-
	STPN [Nguyen *et al.*, 2018]	52.0	44.7	35.5	25.8	16.9	9.9	4.3
	Autoloc [Shou *et al.*, 2018]	-	-	35.8	29.0	21.2	13.4	5.8
	W-TALC [Paul *et al.*, 2018]	55.2	49.6	40.1	31.1	22.8	-	7.6
	MAAN [Yuan *et al.*, 2019]	59.8	50.8	41.1	30.6	20.3	12.0	6.9
	CMCS [Liu *et al.*, 2019]	57.4	50.8	41.2	32.1	23.1	15.0	7.0
	3C-Net [Narayan *et al.*, 2019]	59.1	53.5	44.2	34.1	26.6	-	8.1
	BM [Nguyen *et al.*, 2019]	60.4	56.0	46.6	37.5	26.8	17.6	9.0
	BaSNet [Lee *et al.*, 2020]	58.2	52.3	44.6	36.0	27.0	18.6	10.4
	DGAM [Shi *et al.*, 2020]	60.0	54.2	46.8	38.2	28.8	19.8	**11.5**
	ActionBytes [Jain *et al.*, 2020]	-	-	43.0	35.8	29.0	-	9.5
	ACL [Gong *et al.*, 2020]	-	-	46.9	38.9	30.1	19.8	10.4
	TSCN [Zhai *et al.*, 2020]	63.4	57.6	47.8	37.7	28.7	19.4	10.2
	A2CL-PT [Min and Corso, 2020]	61.2	56.1	48.1	39.0	30.1	19.2	10.6
	Ours	**64.8**	**58.4**	**50.8**	**42.2**	**32.9**	**21.0**	10.1

Table 1: Comparisons on the THUMOS14 test set for fully-supervised and weakly-supervised temporal action localization.

time, e_i is the end time, c is the action category and q_i is the proposal score. As in [Liu *et al.*, 2019], q_i is calculated by:

$$q_i = mean(S[inner, c]) - mean(S[outer, c]) + \gamma p_c, \quad (9)$$

where $inner$ denotes the region (b_i, e_i), and $outer$ denotes the surrounding region $(b_i - (e_i - b_i)/4, b_i) \cup (e_i, e_i + (e_i - b_i)/4)$. The category probability is combined with the weight γ. Non-maximum suppression(NMS) is used to remove duplicate proposals and generate the final localization results.

4 Evaluations

4.1 Data Description and Evaluation Protocol

To evaluate our method, we conduct experiments on two video benchmarks: THUMOS14 [Idrees *et al.*, 2017] and ActivityNet [Heilbron *et al.*, 2015].

THUMOS14. It contains untrimmed videos with temporal annotations from 20 action classes. There are 200 videos in the validation set and 212 videos in the testing set. Following the settings in previous works [Chao *et al.*, 2018; Liu *et al.*, 2019], we train our model on the validation set and evaluate on the test set.

ActivityNet. To facilitate comparisons, we conduct experiments on both ActivityNet-1.2 and ActivityNet-1.3. ActivityNet-1.3 contains about 20,000 videos from 200 activity classes. ActivityNet-1.2 is a subset of ActivityNet-1.3, which has videos from 100 activity classes. This dataset is divided into training, validation and testing sets with a ratio of 2:1:1. Since the annotations of testing set are withheld, we use the training set to train our model and evaluate on the validation set as in previous work [Paul *et al.*, 2018; Narayan *et al.*, 2019].

Evaluation protocol. We report the traditional mean Average Precision (mAP) at different temporal intersection over union (IoU) thresholds. The average mAP with IoU thresholds [0.5:0.95:0.05] is used to compare different methods on ActivityNet. On THUMOS14, the IoU thresholds are from 0.1 to 0.7 with a stride of 0.1.

4.2 Implementation Details

We implement our model in PyTorch. To extract video features, we utilize I3D [Carreira and Zisserman, 2017] models pre-trained on Kinetics [Carreira and Zisserman, 2017]. For each snippet, the corresponding optical flow is generated using the TV-L1 algorithm. The input of I3D is 16 stacked RGB or optical flow frames. The output is a 1024-D feature for each stream. Two separate models are trained for RGB and flow streams respectively. Then outputs of RGB and optical flow streams are combined by late fusion to generate the action localization results.

The action localization model is trained with batch size 24 and optimized by Adam. The learning rate of localization model is 0.001 on ActivityNet and 0.0001 on THUMOS14. The policy network is optimized by Adam with 0.0001 learning rate on ActivityNet and 0.00001 learning rate on THUMOS14. α in Eqn. 1 is 0.5. For action localization, classes whose video-level probabilities below 0.1 are filtered out. For the remaining class c, a set of threshold values ranging from $[0.1 : 1.0 : 0.1] \times mean\,(S[:, c])$ is used to generate action proposals. γ is set to 0.1 when scoring proposals.

4.3 Comparisons with State-of-the-art

Table 1 summarizes comparisons with existing weakly-supervised and fully-supervised methods on THUMOS14. Our method outperforms other weakly-supervised methods when IoU varies from 0.1 to 0.6. Specifically, for mAP@0.5, our method improves the performance by 2.8%. Furthermore, the results achieved by our weakly-supervised method are comparable to the results obtained by several fully-supervised methods, indicating the effectiveness of our method.

Tables 2 and 3 present the results on benchmarks ActivityNet-1.2 and ActivityNet-1.3 respectively. On both

Methods	mAP@IoU (%)			
	0.5	0.75	0.95	AVG
AutoLoc [Shou *et al.*, 2018]	27.3	15.1	3.3	16.0
CMCS [Liu *et al.*, 2019]	36.8	22.0	5.6	22.4
3C-Net [Narayan *et al.*, 2019]	37.2	-	-	21.7
TSM [Yu *et al.*, 2019]	28.3	17.0	3.5	-
CleanNet [Le Wang *et al.*, 2019]	37.1	20.3	5.0	21.6
RPN [Huang *et al.*, 2020]	37.6	23.9	5.4	23.3
BaSNet [Lee *et al.*, 2020]	38.5	24.2	5.6	24.3
ACL [Gong *et al.*, 2020]	40.0	25.0	4.6	24.6
TSCN [Zhai *et al.*, 2020]	37.6	23.7	**5.7**	23.6
EM-MIL [Luo *et al.*, 2020]	37.4	-	-	-
Ours	**45.5**	**27.3**	5.4	**27.6**

Table 2: Comparisons on the ActivityNet1.2 dataset for action localization. AVG denotes average mAP on thresholds 0.5:0.05:0.95

Methods	mAP@IoU (%)			
	0.5	0.75	0.95	AVG
CMCS [Liu *et al.*, 2019]	34.0	20.9	**5.7**	21.2
BaSNet [Lee *et al.*, 2020]	34.5	22.5	4.9	22.2
MAAN [Yuan *et al.*, 2019]	33.7	21.9	5.5	-
BM [Nguyen *et al.*, 2019]	36.4	19.2	2.9	-
TSCN [Zhai *et al.*, 2020]	35.3	21.4	5.3	21.7
A2CL-PT [Min and Corso, 2020]	36.8	22.0	5.2	22.5
Ours	**41.8**	**26.2**	5.0	**26.0**

Table 3: Comparisons on the ActivityNet1.3 dataset for action localization.AVG denotes average mAP on thresholds 0.5:0.05:0.95

versions of ActivityNet, our method significantly outperforms other state-of-the-art weakly-supervised methods in terms of average mAP. The performance of our method is slightly lower than other methods when IoU=0.95. Since the difference between actions surrounding context and action instances is small, temporal boundaries of actions have an intrinsic ambiguity. We observe that our method sometimes generates overly complete proposals containing surrounding context, leading to false positives when IoU is high. Nonetheless, we improve the average mAP on ActivityNet-1.3 from the previous state-of-the-art 22.5% to 26.0%.

4.4 Ablation Studies

To analyze how each component contributes to the overall performance, we conduct ablation studies on the THUMOS14 test set. We start with an action localization model without Siamese architecture and self-refine module. The model is trained only by $\mathcal{L}_{cls}$. Adding $\mathcal{L}_{trans\_cls}$ indicates the Siamese architecture is used, and the model is supervised by the classification loss of original and transformed videos. The self-supervised consistency loss $\mathcal{L}_{cons}$ is further included to validate the effectiveness of self-supervision. Finally, the self-refine module is added to get our full model.

Table 4 demonstrates the results by considering one more component at each stage. Adding $\mathcal{L}_{trans\_cls}$ to the baseline model improves the mAP by 1.6%. Self-supervision consistency loss contributes a significant increase of 4.6%, demonstrating the effectiveness of the proposed equivariant constraint. The self-refine module further improves the mAP from 28.8% to 32.9%, showing that temporal and semantic context is critical to generate accurate CAS.

$\mathcal{L}_{cls}$	$\mathcal{L}_{trans\_cls}$	$\mathcal{L}_{cons}$	SR	mAP@0.5
✓	-	-	-	22.6
✓	✓	-	-	24.2
✓	✓	✓	-	28.8
✓	✓	✓	✓	32.9

Table 4: Contribution of each design in our model on THUMOS14 test set. SR means self-refine module.

Type	Operation	mAP@0.5
Single	Resize	27.8
	Window Warp	29.1
	Attention Based Time Warp	31.1
All	Random Select	31.3
	Adversarial Select	32.9

Table 5: Ablation studies results on THUMOS14 test set for different temporal transforms.

Table 5 shows the results of different temporal transform operations. To validate the effectiveness of adversarial temporal transformation, we compare the following five settings: 1) only use resize type of transform operations (Resize); 2) only use window warp type of transform operations (Window Warp); 3) only use attention based time warp type of transform operations (Attention Based Time Warp); 4) use all three type of transform operations, and randomly select one operation from all operations at each training step (Random Select); 5) use all three type of transform operations. The policy network is used to select one operation from all operations at each training step (Adversarial Select). The result shows that when single type of operation is used, the performance of attention based time warp methods is better than other two types, which proves its effectiveness. In specific, compared to Resize operation, attention based time warp improves the mAP from 27.8% to 31.1%. Compared with the random select, adversarial select improves the mAP by 1.6%, verifying the effectiveness of policy network. When using all types of operations, the performance is better than using single type of operation, indicating different types of operations are complementary.

5 Conclusions

This paper explores self-supervised equivariant consistency regulation for weakly-supervised temporal action localization. We design a set of temporal transform operations and utilize a policy network to select an operation in an adversarial manner. On THUMOS14 and ActivityNet, our methods re-calibrate the state-of-the-are performance on weakly-supervised temporal action localization.

Acknowledgements

This work is supported by National Natural Science Foundation of China (61772037), Beijing Natural Science Foundation (Z190001) and Tencent AI Lab Rhino-Bird Focused Research Program (JR202021).

References

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

[Chao *et al.*, 2018] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster R-CNN architecture for temporal action localization. In *CVPR*, 2018.

[Doersch *et al.*, 2015] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[Gidaris *et al.*, 2018] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[Gong *et al.*, 2020] Guoqiang Gong, Xinghan Wang, Yadong Mu, and Qi Tian. Learning temporal co-attention models for unsupervised video action localization. In *CVPR*, 2020.

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

[Huang *et al.*, 2020] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Relational prototypical network for weakly supervised temporal action localization. In *AAAI*, 2020.

[Idrees *et al.*, 2017] Haroon Idrees, Amir Roshan Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[Isola *et al.*, 2015] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H. Adelson. Learning visual groups from co-occurrences in space and time. *CoRR*, 2015.

[Jain *et al.*, 2020] Mihir Jain, Amir Ghodrati, and Cees G. M. Snoek. Actionbytes: Learning from trimmed videos to localize actions. In *CVPR*, 2020.

[Le Wang *et al.*, 2019] 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.

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

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

[Luo *et al.*, 2020] 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. In *ECCV*, 2020.

[Min and Corso, 2020] Kyle Min and Jason J. Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. In *ECCV*, 2020.

[Misra *et al.*, 2016] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016.

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

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

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

[Noroozi *et al.*, 2017] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.

[Paul *et al.*, 2018] Sujoy Paul, Sourya Roy, and Amit K. Roy-Chowdhury. W-TALC: weakly-supervised temporal activity localization and classification. In *ECCV*, 2018.

[Shi *et al.*, 2020] Baifeng Shi, Qi Dai, Yadong Mu, and Jingdong Wang. Weakly-supervised action localization by generative attention modeling. In *CVPR*, 2020.

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

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

[Wang *et al.*, 2017b] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *ICCV*, 2017.

[Wang *et al.*, 2020] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020.

[Williams, 1992] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992.

[Xu *et al.*, 2020] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali K. Thabet, and Bernard Ghanem. G-TAD: sub-graph localization for temporal action detection. In *CVPR*, 2020.

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

[Yuan *et al.*, 2019] Yuan Yuan, Yueming Lyu, Xi Shen, Ivor W. Tsang, and Dit-Yan Yeung. Marginalized average attentional network for weakly-supervised learning. In *ICLR*, 2019.

[Zhai *et al.*, 2020] Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, Junsong Yuan, and Gang Hua. Two-stream consensus network for weakly-supervised temporal action localization. In *ECCV*, 2020.

[Zhang *et al.*, 2016] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, 2016.

[Zhao *et al.*, 2017] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.