HR-Pro: Point-Supervised Temporal Action Localization via Hierarchical Reliability Propagation

Huaxin Zhang[1], **Xiang Wang**[1], **Xiaohao Xu**[2], **Zhiwu Qing**[1], **Changxin Gao**[1], **Nong Sang**[1]*

[1] Key Laboratory of Image Processing and Intelligent Control,
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2] University of Michigan, Ann Arbor
{zhanghuaxin, wxiang, qzw, cgao, nsang}@hust.edu.cn, {xiaohaox}@umich.edu

Abstract

Point-supervised Temporal Action Localization (PSTAL) is an emerging research direction for label-efficient learning. However, current methods mainly focus on optimizing the network either at the snippet-level or the instance-level, neglecting the inherent reliability of point annotations at both levels. In this paper, we propose a **H**ierarchical **R**eliability **Pro**pagation (**HR-Pro**) framework, which consists of two reliability-aware stages: Snippet-level Discrimination Learning and Instance-level Completeness Learning, both stages explore the efficient propagation of high-confidence cues in point annotations. For snippet-level learning, we introduce an online-updated memory to store reliable snippet prototypes for each class. We then employ a Reliability-aware Attention Block to capture both intra-video and inter-video dependencies of snippets, resulting in more discriminative and robust snippet representation. For instance-level learning, we propose a point-based proposal generation approach as a means of connecting snippets and instances, which produces high-confidence proposals for further optimization at the instance level. Through multi-level reliability-aware learning, we obtain more reliable confidence scores and more accurate temporal boundaries of predicted proposals. Our HR-Pro achieves state-of-the-art performance on multiple challenging benchmarks, including an impressive average mAP of ***60.3%*** on THUMOS14. Notably, our HR-Pro largely surpasses all previous point-supervised methods, and even outperforms several competitive fully-supervised methods. Code will be available at https://github.com/pipixin321/HR-Pro.

Introduction

Temporal action localization is a fundamental task in video understanding field, which attempts to temporally localize and classify action instances in the untrimmed video, and has attracted increasing attention due to its potential application in various fields (Lee, Ghosh, and Grauman 2012; Vishwakarma and Agrawal 2013). However, traditional fully-supervised methods (Lin et al. 2018, 2019; Xu et al. 2020; Qing et al. 2021; Wang et al. 2022b,a; Nag et al. 2022) require accurate temporal annotations, which are extremely time-consuming and labor-demanding, hindering the practical applications. Therefore, many researchers (Wang et al. 2017; Shou et al. 2018; Wang et al. 2021b) start to pay attention to weakly-supervised temporal action localization (WSTAL) where only video-level labels are available. Although significant progress in WSTAL has been made, the lack of action boundary information imposes a great challenge for models to distinguish actions and backgrounds, resulting in unsatisfactory performance compared to fully-supervised methods.

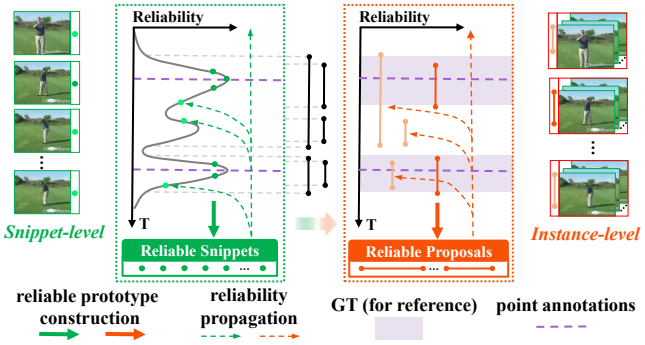

Figure 1: Motivation illustration. Given the point-level annotation (in purple), we consider the action reliability prior both at the snippet level and instance proposal level to enable reliability-aware action representation learning. In specific, our insight is to propagate reliable prototypes to produce more discriminative snippet-level scores and more reliable and complete instance-level scores. Darker color (greener or more orange) indicates higher reliability. Here, a case with one action class is shown for brevity.

Under the WSTAL setting, to balance labeling cost and detection performance, Ma *et al*. (Ma et al. 2020) introduce the point-supervised temporal action localization (PSTAL) task which provides only a timestamp label for each action instance. Their pioneering research indicates that point-level annotations consume almost comparable labor costs as video-level annotations while providing richer guidance information. Subsequently, many works start to follow this setting and propose various customized solutions. Typically, LACP (Lee and Byun 2021) proposes to learn completeness of action instances by searching the optimal pseudo sequence with a greedy algorithm. Ju *et al*. (Ju et al. 2021) proposes a seed frame detector to generate proposals and

*indicates corresponding author.

then performs regression and classification on the proposals.

Although previous methods have achieved impressive results, they are still limited to optimizing the network at either the snippet level or instance level. Snippet-level approaches (Ma et al. 2020; Lee and Byun 2021) may produce many unreliable (*e.g.*, overcomplete or false positive) detections because they only consider individual snippets, which ignore the overall action instance. On the other hand, instance-level approach (Ju et al. 2021) cannot achieve satisfactory optimization due to the absence of reliable proposals generated from snippet scores. We propose that the high reliability of point annotations can be propagated at both snippet and instance levels. To accomplish this, we derive reliable prototypes at different levels by considering their confidence scores and relative positions to point annotations. Leveraging these reliable prototypes through high-confidence information propagation enables the network to learn more discriminative snippet-level representation and more reliable instance-level proposals.

Building upon these insights, we present a Hierarchical Reliability Propagation method that consists of two reliability-aware stages: Snippet-level Action Discrimination Learning and Instance-level Action Completeness Learning. These stages are illustrated in Fig. 1. (1) In the Snippet-level Action Discrimination Learning stage, our objective is to obtain discriminative snippet-level scores for generating more reliable proposals. To achieve this, we introduce an online-updated memory to store reliable prototypes for each class. Additionally, we propose a Reliability-aware Attention Block to propagate high-confidence cues from these reliable prototypes to other snippets. Through contrastive optimization of the memory and snippet features, we derive a more discriminative action representation. (2) In the Instance-level Action Completeness Learning stage, we refine the confidence score and boundary of the proposals through instance-level feature learning. We propose a point-based proposal generation method that produces reliable instance-level prototype proposals, along with high-confidence positive and negative proposals. These proposals' features are then fed into a Score Head and a Regression Head to predict the completeness score and refined boundary. This prediction process is guided by reliable instance prototypes. As a result, the network can estimate more reliable instance-level scores and achieve more accurate temporal boundaries.

To summarize, our contributions are as follows:

- Our proposed method, *i.e.*, HR-Pro, is the first to leverage inherent reliability of point annotations at both snippet and instance level optimization in the PSTAL domain.
- At the snippet level, we propose a reliability-aware attention module and reliable-memory-based contrastive loss to acquire discriminative snippet-level representation.
- At the instance level, we propose reliability-based proposal generation and ranking method to produces high-confidence proposals for further optimization at the instance level.
- Our HR-Pro achieves state-of-the-art performance on four standard temporal action localization benchmarks, including an impressive average mAP of ***60.3%*** on THUMOS14, **which even surpasses several competitive fully-supervised methods**.

Related Work

Fully-supervised temporal action localization. Mainstream fully-supervised methods can be divided into two categories, *i.e.*, one-stage and two-stage. The one-stage methods (Xu et al. 2020; Zhang, Wu, and Li 2022) simultaneously predicts the boundary and category of action as the final detection result. The two-stage methods (Lin et al. 2018, 2019; Qing et al. 2021; Wang et al. 2022b, 2021a) first generate numerous proposals and then classify the proposals. Despite the significant progress in recent years, these fully-supervised methods require expensive annotation costs, which limits their application.

Weakly-supervised temporal action localization. To reduce the labeling cost, many weakly-supervised temporal action localization (WSTAL) methods (Wang et al. 2017; Shou et al. 2018; Liu et al. 2019) have been proposed where only video-level labels are available. Most recent WSTAL methods follow the localization-by-classification mode. They first use a snippets classifier to evaluate the class probability of each video snippet, *i.e.*, Class Activation Sequence (CAS), and then locate the temporal boundary using multiple predefined thresholds. Recently, many attempts have been made to enhance the performance of the model. BaS-Net (Lee, Uh, and Byun 2020) introduces background classes and background branch to suppress class activation values of background snippets. ACM-Net (Qu et al. 2021) proposes three attention branches to separate foreground, background, and context. CoLA (Zhang et al. 2021) proposes a hard snippet mining algorithm and a snippet contrastive loss to refine the hard snippet representation in feature space. ACG-Net (Yang, Qin, and Huang 2022) and DGCNN (Shi et al. 2022) adopt graph networks to enhance feature embedding and model relationships between action snippets. ASM-Loc (He et al. 2022) proposes to use intra- and inter-segment attention for modeling action dynamics and capturing temporal dependencies. Due to the absence of frame-wise annotations, the perfomance of these models falls largely behind the fully-supervised methods.

Point-supervised temporal action localization. To balance labeling cost and model performance, point-supervised temporal action localization (PSTAL) task is proposed by (Ma et al. 2020), which provides a timestamp label for each action instance. To explore the guidance information provided by point annotations, SF-Net (Ma et al. 2020) uses the single-frame label to mine its adjacent pseudo label for training classifiers. Ju et.al (Ju et al. 2021) uses a two-stage approach, which proposes a seed frame detector to generate proposals and then performs regression and classification on the proposals. LACP (Lee and Byun 2021) searches the optimal pseudo sequence through a greedy algorithm which is used to guide the network to learn the completeness of action instances. However, these methods are limited in optimizing the network either at snippet-level or at instance-level, leading to less effective discriminative representations at the snippet-level and unreliable scores at the instance-level.

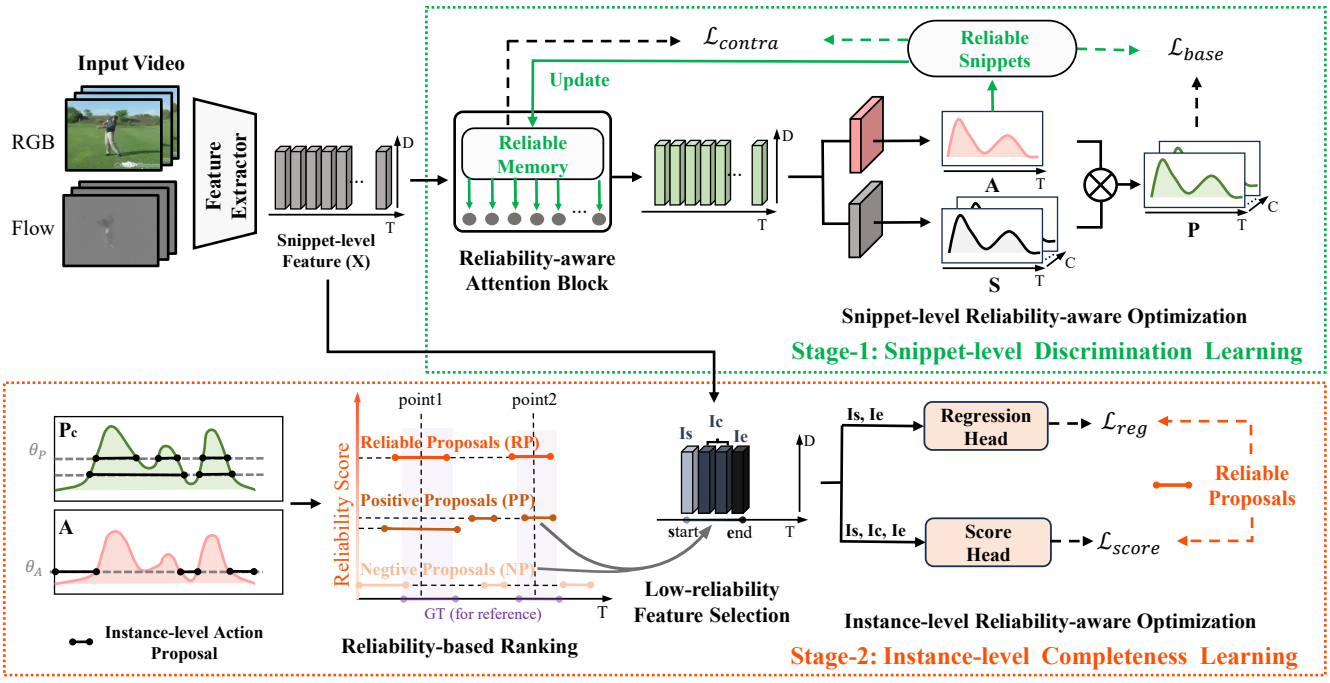

Figure 2: Overview of Hierarchical Reliability Propagation (HR-Pro). We propagate reliable prototypes during two-stage action localization learning, *i.e.*, Snippet-level Discrimination Learning and Instance-level Completeness Learning. (1) *Snippet* level: we aim to obtain snippet representations with good inter-class discrimination and action-background discrimination. (2) *Instance* level: we aim to refine the confidence score and boundary of the coarse proposals generated from snippet-level output.

Preliminaries

Problem Definition. For point-supervised temporal action localization (PSTAL), models are trained with a single-frame annotation for each untrimmed video. Each action instance is annotated with a temporal point p_i and a one-hot vector y_{p_i} indicating the action category c with $y_{p_i}[c] = 1$. The video contains a total of N action instances. During inference, we generate predicted results for each test video using $(s_m, e_m, c_m, p_m)_m^M$, where s_m and e_m are the start and end times of the m-th predicted action instance, c_m is the predicted category, and p_m is the confidence score. M is the total number of predicted action instances.

Baseline Architecture. The input video is first divided into multi-frame snippets, then we use a pre-trained video classification model to extract RGB and optical flow features of each snippet and concatenate them along the channel dimension. The features of the input video are formulated as $\mathbf{X} \in \mathbb{R}^{T \times D}$, where T and D indicate the number of snippets and the dimension of features, respectively. The features are then fed into a feature embedding layer get task-specific embedded features $\mathbf{X}^e \in \mathbb{R}^{T \times D}$.

Following previous work (Lee and Byun 2021), we first input the embedded features into a snippet-level classifier to obtain class-specific activation sequence (CAS) $\mathbf{S} \in \mathbb{R}^{T \times C}$, where C denotes the class number. To reduce the noise from background snippets, we use a convolutional layer to generate a class-agnostic attention sequence $\mathbf{A} \in \mathbb{R}^T$. Then, we fuse them by element-wise production to get the final snippet-level predictions $\mathbf{P} \in \mathbb{R}^{T \times C}$ where $\mathbf{P} = \mathbf{S} \cdot \mathbf{A}$.

Baseline Optimization Loss. Based on the characteristic that each action instance contains a point annotation and the adjacent point annotations are in different action instances, we select pseudo action snippets $\mathcal{T}^+ = \{t_i\}_{i=1}^{N_{act}}$ and background snippets $\mathcal{T}^- = \{t_j\}_{j=1}^{N_{bkg}}$ based on point annotations and class-agnostic attention sequence. Specifically, snippets near a point annotation with higher class-agnostic attention than a given threshold are labeled as pseudo-action snippets, which share the same action category as the point annotation. Conversely, snippets located between two adjacent point annotations with the lowest class-agnostic attention or lower class-agnostic attention than a given threshold are labeled as pseudo-background snippets. We use these pseudo snippet samples for supervision:

$$\mathcal{L}_{base} = \frac{1}{N_{act}} \sum_{c=1}^{C} \sum_{t \in \mathcal{T}^+}^{N_{act}} FL(\mathbf{P}_{t,c}) + \frac{1}{N_{bkg}} \sum_{t \in \mathcal{T}^-}^{N_{bkg}} FL(1 - \mathbf{A}_t) \tag{1}$$

where N_{act} and N_{bkg} is the total number of pseudo action snippets and background snippets respectively, FL represents the focal loss function (Lin et al. 2017).

Method: Hierarchical Reliability Propagation

Reliability can help the network mine more pseudo samples, which can alleviate the sparsity of guidance in point-supervised setting. We argue that the inherent reliability of point annotations can be propagated at both snippet

and instance level optimization. Therefore, we propose a Hierarchical Reliability Propagation framework, which divides action localization learning into two cascaded stages: (1) Snippet-level Action Discrimination Learning and (2) Instance-level Action Completeness Learning.

Snippet-level Action Discrimination Learning

Previous works have primarily focused on estimating temporal pseudo-labels to expand training samples, which restricts the propagation of high-confidence snippet information within a single video. Thus, we introduce Reliability-aware Snippet-level Action Discrimination Learning, which proposes to store the reliable prototypes for each class and propagate high-confidence cues from these prototypes to other snippets via intra-video and inter-video ways.

Reliable Prototype Construction. As the snippet-level action representation, *i.e.*, snippet features, only captures short-term and partial action states, the feature can be noisy and unreliable. Thus, our insight is to construct reliable snippet prototypes via a de-noising mechanism for further reliability-guided optimization.

Specifically, we create an online-updated prototype memory to store reliable prototypes for each class during representation learning, enabling us to leverage the feature information from the entire dataset to mitigate the noise of each feature. Formally, we denote the item in memory by $\mathbf{m_c} \in \mathbb{R}^D (c = 1, 2, ..., C)$. Under the PSTAL setting, we initialize the prototype pool by selecting features with point annotations for each class. This is done by computing the average of snippet features x_{p_i} corresponding to the point annotations p_i for class c. We normalize the sum by the total number of point annotations N_c for class c across all training videos. The initial prototype memory is defined as:

$$\mathbf{m}_c^0 = \frac{1}{N_c} \sum_{i}^{N_c} x_{p_i} (y_{p_i}[c] = 1) \tag{2}$$

Next, we update the prototypes for each class using the features of pseudo-action snippets, which is formulated as:

$$\mathbf{m}_c^t = \mu \mathbf{m}_c^{(t-1)} + (1 - \mu) \mathbf{x}_{t_i}^{(t)} \tag{3}$$

Here, μ denotes the momentum coefficient for an update.

As is shown in Fig. 3, to derive the prototype, we input the snippet-level features extracted by the feature extractor into a Reliability-aware Attention Block (RAB). The RAB is specifically designed to capture both intra-video and inter-video dependencies of snippets, enabling the modeling of complementary temporal relationships. Long-term temporal dependency modeling is crucial for long videos, as supported by previous works (Zhang, Wu, and Li 2022; Wang et al. 2022c; Xu et al. 2022; Wang et al. 2023). However, attention tends to become sparse and focus mainly on discriminative snippets within the same video, resulting in limited information interaction. Therefore, the RAB incorporates the insight of propagating global class information from a reliable prototype (*i.e.*, snippet) memory, thereby enhancing the robustness of snippet features and increasing the attention on less discriminative snippets.

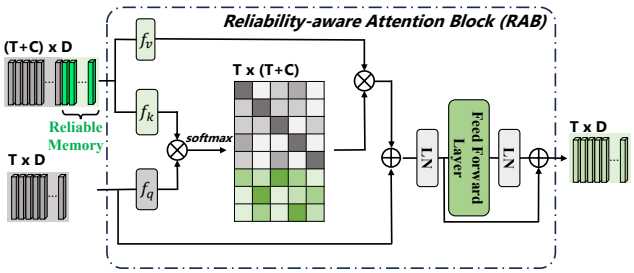

Figure 3: Architecture detail of Reliability-aware Attention Block (RAB). Reliable prototype memory (in green) is injected into the original snippet features (in grey) to introduce reliable cues via the attention mechanism.

Technically, we employ a linear layer f_q to project the video features onto the corresponding query. Subsequently, we concatenate ($[;]$ denotes concatenation) the video features $\mathbf{X}$ with the prototype features $\mathbf{m}_i$ stored in the reliable memory bank. Then, we use separate linear layers, f_k and f_v, to project the concatenated features into key and value, respectively:

$$\mathbf{Q} = f_q(\mathbf{X}) \tag{4}$$

$$\mathbf{K} = f_k([\mathbf{X}; \mathbf{m}_1; ...; \mathbf{m}_C]), \mathbf{V} = f_v([\mathbf{X}; \mathbf{m}_1; ...; \mathbf{m}_C]) \tag{5}$$

Next, we multiply the query and the transposed key to obtain the non-local attention $\mathbf{attn} \in \mathbb{R}^{T \times (T+C)}$.

$$\mathbf{attn} = softmax(\mathbf{Q} \cdot \mathbf{K}^T / \sqrt{D}) \tag{6}$$

Furthermore, we multiply the attention with the value, and pass it through a Feed Forward Layer (FFL) composed of cascaded FC-GELU-FC layers. Here, LayerNorm (LN) and residual connection are used for the normalization and retention of the original information. The output reliability-aware features are fed into subsequent network layers for temporal action localization.

Reliability-aware Optimization. To push away the features of pseudo-action snippets and prototypes from different classes in the reliable prototype pool and push away the features of pseudo-action and background snippets in the same video, we follow a contrastive learning manner and propose a reliability-aware snippet-contrastive loss ($\mathcal{L}_{contra}$):

$$\mathcal{L}_{contra} = -\frac{1}{C} \sum_{c=1}^{C} \sum_{t_i^c} log(\frac{s(\mathbf{x}_{t_i^c}, \mathbf{m}_c)}{s(\mathbf{x}_{t_i^c}, \mathbf{m}_c) + \sum_{\forall k \neq c} s(\mathbf{x}_{t_i^c}, \mathbf{m}_k)} + \frac{s(\mathbf{x}_{t_i^c}, \mathbf{m}_c)}{s(\mathbf{x}_{t_i^c}, \mathbf{m}_c) + \sum_{\forall t_j \in \mathcal{T}^-} s(\mathbf{x}_{t_j}, \mathbf{m}_c)}) \tag{7}$$

where t_i^c indicates the pseudo action snippet of class c; $s(\cdot, \cdot)$ is the similarity function formulated as $s(\mathbf{x_1}, \mathbf{x_2}) = exp(\bar{\mathbf{x}}_1 \cdot \bar{\mathbf{x}}_2 / \tau)$ with a temperature parameter τ, $\bar{\mathbf{x}}$ represent the normalized features of $\mathbf{x}$,

Finally, the overall training objective for Snippet-level Action Discrimination Learning includes both the baseline loss $\mathcal{L}_{base}$ and our reliability-aware snippet optimization loss $\mathcal{L}_{contra}$ weighted by a parameter λ_1:

$$\mathcal{L}_{snippet} = \mathcal{L}_{base} + \lambda_1 \mathcal{L}_{contra} \tag{8}$$

Instance-level Action Completeness Learning

Snippet-level Representation Learning empowers our model with robust snippet-level action discrimination capabilities. However, a snippet-level-based pipeline can produce numerous unsatisfactory detections despite discriminative snippet representation, because the proposal score is unreliable without considering the whole instance (*e.g.*, *running* in background frames has a high snippet score in the *long jump* category, but it is not a complete *long jump* action). To fully explore the temporal structure of actions at the instance level and optimize the score ranking of proposals, we introduce Instance-level Action Completeness Learning. This method is aimed at refining the proposals' confidence score and boundary via instance-level feature learning with the guidance of reliable instance prototypes.

Reliable Prototype Construction. To leverage instance-level priors of point annotations during training, we propose a point-based proposal generation method that yields reliable instance-level prototype proposals, along with high-confidence positive and negative proposals. Initially, we produce candidate proposals for each predicted class by selecting snippets with class-specific activation scores higher than the threshold θ_P. (we use multiple thresholds in implementation.) The OIC (outer-inner-contrast) score (Shou et al. 2018) is calculated for each candidate proposal to gauge its reliability score, represented as p_{OIC}. Lower reliability scores indicate incomplete or over-complete predictions. We formulate each candidate proposal as (s_i, e_i, c_i, p_{OIC}), these proposals are then ranked into two types based on their reliability score and temporal position: (1) Reliable Proposals ($\mathbf{RP}$): for each point in each class, the proposal contained this point, and has the highest reliability (i.e., OIC score); (2) Positive Proposals ($\mathbf{PP}$): all remaining candidate proposals. To ensure a balanced number of positive and negative samples, we group snippets with class-agnostic attention scores lower than pre-defined threshold θ_A, which derives Negative Proposals ($\mathbf{NP}$).

Reliability-aware Optimization. For each proposal, we select all snippet features within the proposal region as its center features I_c, then we expand the boundary of the proposal with ratio ε to get starting region and ending region, which derives starting feature I_s and ending features I_e of the proposal, ε is set to 0.25 in practice.

(1) To predict the completeness score of each proposal, we use the boundary-sensitive proposal features following (Ren et al. 2023) as input to the Score Head ϕ_s.

$$\hat{p}_{comp} = \phi_s([\overline{I}_c - \overline{I}_s; \overline{I}_c; \overline{I}_c - \overline{I}_e]) \tag{9}$$

where $\overline{I}_s, \overline{I}_c, \overline{I}_e$ is the max-pooling feature of I_s, I_c, and I_e along the temporal dimension, respectively.

Then, the reliability-aware supervision for instance level completeness score can be formulated as:

$$\mathcal{L}_{score} = \frac{1}{N_p + N_n} \sum_{i=1}^{N_p+N_n} SmoothL1(\hat{p}_{comp}, g_{comp}) \tag{10}$$

where N_p, N_n are the total number of positive proposals and negative proposals, respectively, g_{comp} represents the Intersection over Union (IoU) between the proposal and the most reliable proposal ($\mathbf{RP}$) that matches it.

(2) To obtain more accurate action proposal boundaries, we input the starting features and ending features of each proposal in $\mathbf{PP}$ into the Regression Head ϕ_r to predict the offset of the start time and the end time, *i.e.*, $\Delta\hat{s}$ and $\Delta\hat{e}$,

$$\{\Delta\hat{s}, \Delta\hat{e}\} = \phi_r([I_s; I_e]) \tag{11}$$

then, the refined proposal can be obtained:

$$\hat{s}_r = s_p - \Delta\hat{s}w_p, \qquad \hat{e}_r = e_p - \Delta\hat{e}w_p \tag{12}$$

where $w_p = e_p - s_p$ is the length of the proposal.

Then, the reliability-aware supervision for instance level boundary regression can be formulated as:

$$\mathcal{L}_{reg} = \frac{1}{N_p} \sum_{i=1}^{N_p} SmoothL1(\hat{r}_{comp}, 1) \tag{13}$$

where $\hat{r}_{comp}$ represents the IoU between the refined proposal and the most reliable proposal ($\mathbf{RP}$) that matches it.

Finally, the reliability-aware instance-level completeness learning has an overall objective function that consists of both regression and score losses weighted by a parameter λ_2, formulated as:

$$\mathcal{L}_{instance} = \mathcal{L}_{score} + \lambda_2 \mathcal{L}_{reg} \tag{14}$$

Temporal Action Localization Inference

We first extract the snippet-level prediction of predicted class P_c and class-agnostic attention A of each video, which is used to generate candidate proposals,represented as (s_i, e_i, c_i, p_{OIC}). Then, we input the instance-level feature of each proposal to the Score and Regression heads, which derives two parts of predicted proposals: the score refined part $(s_i, e_i, c_i, p_{OIC} + \hat{p}_{comp})$ and the boundary refined part $(\hat{s_r}, \hat{e}_r, c_i, p_{OIC} + \hat{p}_r)$, $\hat{p}_r$ is the completeness score of refined proposal estimated by the trained Score Head. Finally, we combine them and employ class-wise soft-NMS (Bodla et al. 2017) to remove duplicate proposals.

Experiments

Experimental Setup

Datasets. We conduct our experiments on four popular action localization datasets, with only point-level annotations used for training. In our experiments, we utilize the point-level annotations provided in (Lee and Byun 2021) for fair comparison. (1) **THUMOS14** (Idrees et al. 2017) provides 413 untrimmed sports videos for 20 action categories, including 200 videos for training and 213 videos for testing, and each video contains an average of 15 action instances. Action instance lengths and video lengths vary widely, which makes this dataset challenging. (2) **GTEA** (Fathi, Ren, and Rehg 2011) provides 28 videos of 7 fine-grained daily activities in a kitchen. Four different subjects perform an activity, and each video contains about 1,800 frames. (3) **BEOID** (Damen et al. 2014) provides 58 video samples with 30 action classes with an average duration of $60s$. There is

Supervision	Method	mAP@IoU (%)							AVG (0.1:0.5)	AVG (0.3:0.7)	AVG (0.1:0.7)
		0.1	0.2	0.3	0.4	0.5	0.6	0.7			
Frame-level (Full)	P-GCN (ICCV'19)	69.5	67.8	63.6	57.8	49.1	-	-	61.6	-	-
	TCANet (CVPR'21)	-	-	60.6	53.2	44.6	36.8	26.7	-	44.3	-
	AFSD (CVPR'21)	-	-	**67.3**	**62.4**	**55.5**	**43.7**	**31.1**	-	**52.0**	-
Video-level (Weak)	CoLA (CVPR'21)	66.2	59.5	51.5	41.9	32.2	22.0	13.1	50.3	32.1	40.9
	TS-PCA (CVPR'21)	67.6	61.1	53.4	43.4	34.3	24.7	13.7	52.0	33.9	42.6
	UGCT (CVPR'21)	69.2	62.9	55.5	46.5	35.9	23.8	11.4	54.0	34.6	43.6
	FAC-Net (ICCV'21)	67.6	62.1	52.6	44.3	33.4	22.5	12.7	52.0	33.1	42.2
	ACG-Net (AAAI'22)	68.1	62.6	53.1	44.6	34.7	22.6	12.0	52.6	33.4	42.5
	RSKP (CVPR'22)	71.3	65.3	55.8	47.5	38.2	25.4	12.5	55.6	35.9	45.1
	DELU (ECCV'22)	71.5	66.2	56.5	47.7	40.5	27.2	**15.3**	56.5	37.4	46.4
	Li *et al*. (CVPR'23)	-	-	56.2	47.8	39.3	**27.5**	15.2	-	37.2	-
	Zhou *et al*.(CVPR'23)	**74.0**	**69.4**	**60.7**	**51.8**	**42.7**	26.2	13.1	**59.7**	**38.9**	**48.3**
Point-level (Weak)	SF-Net (ECCV'20)	68.3	62.3	52.8	42.2	30.5	20.6	12.0	51.2	31.6	41.2
	Ju *et al*. (ICCV'21)	72.3	64.7	58.2	47.1	35.9	23.0	12.8	55.6	35.4	44.9
	LACP (ICCV'21)	75.7	71.4	64.6	56.5	45.3	34.5	21.8	62.7	44.5	52.8
	CRRC-Net (TIP'22)	77.8	73.5	67.1	57.9	46.6	33.7	19.8	64.6	45.1	53.8
	HR-Pro (Ours)	**85.6**	**81.6**	**74.3**	**64.3**	**52.2**	**39.8**	**24.8**	**71.6**$^{\uparrow 7.0}$	**51.1**$^{\uparrow 6.0}$	**60.3**$^{\uparrow 6.5}$

Table 1: Comparisons of detection performance on THUMOS14. We include the methods under video-level and frame-level supervision for reference. We utilize the same annotations under the point-level supervision as (Lee and Byun 2021). ↑ denotes the relative performance gain between our method (the best) and the second-best method under point-level supervision setting.

Method	GTEA					BEOID					ActivityNet1.3			
	mAP@IoU (%)					mAP@IoU (%)					mAP@IoU (%)			
	0.1	0.3	0.5	0.7	AVG[0.1:0.7]	0.1	0.3	0.5	0.7	AVG[0.1:0.7]	0.5	0.75	0.95	AVG[0.5:0.95]
SF-Net (ECCV'20)	58.0	37.9	19.3	11.9	31.0	62.9	40.6	16.7	3.5	30.9	-	-	-	-
Ju *et al*. (ICCV'21)	59.7	38.3	21.9	18.1	33.7	63.2	46.8	20.9	5.8	34.9	-	-	-	-
Li *et al*. (CVPR'21)	60.2	44.7	28.8	12.2	36.4	71.5	40.3	20.3	5.5	34.4	-	-	-	-
LACP (ICCV'21)	63.9	55.7	33.9	**20.8**	43.5	76.9	61.4	42.7	25.1	51.8	40.4	24.6	5.7	25.1
CRRC-Net (TIP'22)	-	-	-	-	-	-	-	-	-	-	39.8	24.1	5.9	24.0
HR-Pro (Ours)	**72.6**	**61.1**	**37.3**	17.5	**47.3**$^{\uparrow 3.8}$	**78.5**	**72.1**	**55.3**	**26.1**	**59.4**$^{\uparrow 7.6}$	**42.8**	**27.2**	**8.0**	**27.1**$^{\uparrow 2.0}$

Table 2: Comparisons of detection performance on GTEA, BEOID, and ActivityNet1.3 datasets.

Snippet-level			Instance-level				**AVG**
$\mathcal{L}_{base}$	$\mathcal{L}_{contra}$	RAB	$\mathcal{L}_{reg}$	$\mathcal{L}_{score}$	RP	NP	[0.1:0.7]
✓							49.4
✓	✓						51.5
✓	✓	✓					54.7$^{\uparrow 5.3}$
✓	✓	✓	✓				56.1
✓	✓	✓		✓			57.1
✓	✓	✓	✓	✓			57.8
✓	✓	✓	✓		✓		56.8
✓	✓	✓		✓	✓		58.1
✓	✓	✓	✓	✓	✓		59.1
✓	✓	✓	✓	✓	✓	✓	**60.3**$^{\uparrow 10.9}$
✓			✓	✓	✓	✓	53.9
✓	✓		✓	✓	✓	✓	56.2
✓	✓	✓	✓	✓	✓	✓	**60.3**$^{\uparrow 10.9}$

Table 3: Ablation study on THUMOS14. ↑ denotes the relative gain between each setting and baseline ($\mathcal{L}_{base}$ only).

an average of 12.5 action instances per video. (4) **ActivityNet 1.3** (Caba Heilbron et al. 2015) provides 10,024 training, 4,926 validation, and 5,044 test videos with 200 action classes. Each video includes 1.6 action instances on average.

Evaluation metric. We follow the standard protocols to evaluate with mean Average Precision (mAP) under different intersection over union (IoU) thresholds. A proposal is regarded as positive only if both IoU exceeds the set threshold and the category prediction is correct.

Implementation Details. For a fair comparison, we follow existing method (Lee and Byun 2021) to divide each video into 16-frame snippets and use two-stream I3D network pretrained on Kinetics-400 (Carreira and Zisserman 2017) as the feature extractor. For THUMOS14, we use the Adam optimizer with a learning rate of 1e-4 and a weight decay of 1e-3, and the batch size is set to 16. The hyper-parameters are set by grid search: $\tau = 0.1$, $\mu = 0.999$, $\lambda_1 = \lambda_2 = 1$. The video-level threshold is set to 0.5, the θ_P spans from 0 to 0.25 with a step size of 0.05, the θ_A spans from 0 to 0.1 with a step size of 0.01. The number of RAB is set to 2.

Comparison with State-of-The-Art Methods

We evaluate the effectiveness of our proposed method by comparing it against the most recent fully-supervised and weakly-supervised temporal action localization methods.

THUMOS14. Our proposed method, *i.e.*, HR-Pro, achieves

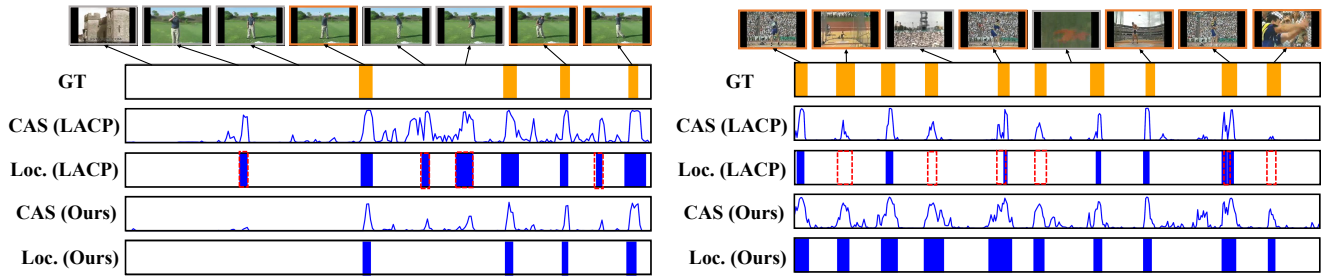

Figure 4: Qualitative results for two action categories, *GolfSwing* (left) and *HammerThrow* (right), on THUMOS14. We compare the detection results of HR-Pro and LACP. The orange and blue bars indicate the ground truth and predicted localization results, respectively; Blue curves represent snippet-level prediction. Prediction errors are bound with red bounding boxes.

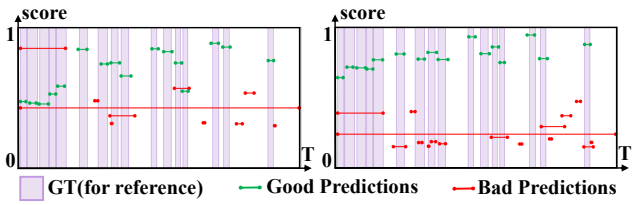

Figure 5: Visualization of detection results on THUMOS14 dataset before (left) and after (right) instance-level completeness learning. The x-axis and y-axis represent time and the reliability score, respectively. We observe that the discrepancy between good and bad predictions is enlarged significantly after instance-level completeness learning.

state-of-the-art performance on THUMOS14 testing set for point-level weakly-supervised temporal action localization. Compared to previous state-of-the-art methods in Table 1, HR-Pro has an average mAP of 60.3% for IoU thresholds of 0.1:0.7, outperforming the prior SoTA method (Fu, Gao, and Xu 2022) by 6.5% for the same thresholds. Notably, our point-supervised method is able to achieve comparable performance with competitive fully-supervised methods, such as AFSD (51.1% vs. 52.0% in average mAP for IoU thresholds of 0.3:0.7). Moreover, HR-Pro demonstrates superior detection performance compared to video-level weakly-supervised methods with similar labeling cost, thanks to the position information provided by point labels.

GTEA & BEOID & ActivityNet 1.3. We demonstrate the generality and superiority of HR-Pro on diverse benchmarks in Table 2. Our method significantly outperforms existing methods, achieving improvements of 3.8%, 7.6%, and 2.0%, on GTEA, BEOID, and ActivityNet 1.3, respectively.

Ablation Study

To further analyze the contribution of model components compared to the baseline setting (with a detection result of 49.4%), we perform a set of ablation studies on THUMOS14. The results are summarized in Table 3.

Snippet-level Discrimination Learning. The introduction of contrastive loss increases performance by 2.1%. Contrastive optimization not only reduces classification errors but also improves the model's ability to distinguish between action and background, thereby improving detection performance. The introduction of Reliability-aware Attention Block (RAB) further improves detection performance by 3.2%. We speculate that the introduction of RAB increases the attention on less reliable action snippets, thus detecting more non-discriminative actions.

Instance-level Completeness Learning. We see the introduction of regression loss and score loss significantly increases the detection performance. The introduction of reliable proposals and negative proposals generated based on point annotations further boosts the results. These results demonstrate that the components of instance-level completeness learning complement each other and make the network estimate the proposal score and boundaries more accurately.

Qualitative Results

Qualitative Comparison. In Fig. 4, we compare our HR-Pro with LACP for temporal action localization on test videos in THUMOS14. Our model shows more accurate detection of action instances. In specific, for *GolfSwing* action, our method effectively distinguishes between action and background snippets, mitigating false action predictions that LACP struggles with; for *HammerThrow* action, our method detects more complete snippets than LACP, which has lower activation values on non-discriminative action snippets.

Effect of Instance-level Completeness Learning. Fig. 5 shows that completeness learning helps our method reduce the score of overcomplete and false positive proposals, leading to improved detection results.

Conclusion

This paper introduces a new framework named HR-Pro for point-supervised temporal action localization. HR-Pro comprises two reliability-aware stages that efficiently propagate high-confidence cues from point annotations at both the snippet and instance level, which enables the network to learn more discriminative snippet representation and more reliable proposals. Extensive experiments on multiple benchmarks demonstrate that HR-Pro significantly outperforms existing methods and achieves state-of-the-art results, which demonstrates the effectiveness of our method and the potential of point annotations.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under grant U22B2053.

References

Bodla, N.; Singh, B.; Chellappa, R.; and Davis, L. S. 2017. Soft-NMS–improving object detection with one line of code. In *ICCV*, 5561–5569.

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.

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

Damen, D.; Leelasawassuk, T.; Haines, O.; Calway, A.; and Mayol-Cuevas, W. W. 2014. You-Do, I-Learn: Discovering Task Relevant Objects and their Modes of Interaction from Multi-User Egocentric Video. In *BMVC*, 3.

Fathi, A.; Ren, X.; and Rehg, J. M. 2011. Learning to recognize objects in egocentric activities. In *CVPR*, 3281–3288.

Fu, J.; Gao, J.; and Xu, C. 2022. Compact Representation and Reliable Classification Learning for Point-Level Weakly-Supervised Action Localization. *IEEE Transactions on Image Processing*, 7363–7377.

He, B.; Yang, X.; Kang, L.; Cheng, Z.; Zhou, X.; and Shrivastava, A. 2022. ASM-Loc: action-aware segment modeling for weakly-supervised temporal action localization. In *CVPR*, 13925–13935.

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". *Computer Vision and Image Understanding*, 1–23.

Ju, C.; Zhao, P.; Chen, S.; Zhang, Y.; Wang, Y.; and Tian, Q. 2021. Divide and conquer for single-frame temporal action localization. In *ICCV*, 13455–13464.

Lee, P.; and Byun, H. 2021. Learning action completeness from points for weakly-supervised temporal action localization. In *ICCV*, 13648–13657.

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

Lee, Y. J.; Ghosh, J.; and Grauman, K. 2012. Discovering important people and objects for egocentric video summarization. In *2012 IEEE conference on computer vision and pattern recognition*, 1346–1353.

Lin, T.; Liu, X.; Li, X.; Ding, E.; and Wen, S. 2019. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, 3889–3898.

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.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, Z.; Wang, L.; Zhang, Q.; Gao, Z.; Niu, Z.; Zheng, N.; and Hua, G. 2019. Weakly supervised temporal action localization through contrast based evaluation networks. In *ICCV*, 3899–3908.

Ma, F.; Zhu, L.; Yang, Y.; Zha, S.; Kundu, G.; Feiszli, M.; and Shou, Z. 2020. Sf-net: Single-frame supervision for temporal action localization. In *ECCV*, 420–437. Springer.

Nag, S.; Zhu, X.; Song, Y.-Z.; and Xiang, T. 2022. Proposal-free temporal action detection via global segmentation mask learning. In *ECCV*, 645–662.

Qing, Z.; Su, H.; Gan, W.; Wang, D.; Wu, W.; Wang, X.; Qiao, Y.; Yan, J.; Gao, C.; and Sang, N. 2021. Temporal context aggregation network for temporal action proposal refinement. In *CVPR*, 485–494.

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*.

Ren, H.; Yang, W.; Zhang, T.; and Zhang, Y. 2023. Proposal-Based Multiple Instance Learning for Weakly-Supervised Temporal Action Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2394–2404.

Shi, H.; Zhang, X.-Y.; Li, C.; Gong, L.; Li, Y.; and Bao, Y. 2022. Dynamic Graph Modeling for Weakly-Supervised Temporal Action Localization. In *ACMMM*, 3820–3828.

Shou, Z.; Gao, H.; Zhang, L.; Miyazawa, K.; and Chang, S.-F. 2018. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 154–171.

Vishwakarma, S.; and Agrawal, A. 2013. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 983–1009.

Wang, L.; Xiong, Y.; Lin, D.; and Van Gool, L. 2017. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 4325–4334.

Wang, Q.; Zhang, Y.; Zheng, Y.; and Pan, P. 2022a. Rcl: Recurrent continuous localization for temporal action detection. In *CVPR*, 13566–13575.

Wang, X.; Qing, Z.; Huang, Z.; Feng, Y.; Zhang, S.; Jiang, J.; Tang, M.; Gao, C.; and Sang, N. 2021a. Proposal relation network for temporal action detection. *arXiv preprint arXiv:2106.11812*.

Wang, X.; Qing, Z.; Huang, Z.; Feng, Y.; Zhang, S.; Jiang, J.; Tang, M.; Shao, Y.; and Sang, N. 2021b. Weakly-supervised temporal action localization through local-global background modeling. *arXiv preprint arXiv:2106.11811*.

Wang, X.; Zhang, H.; Zhang, S.; Gao, C.; Shao, Y.; and Sang, N. 2022b. Context-aware Proposal Network for Temporal Action Detection. *arXiv preprint arXiv:2206.09082*.

Wang, X.; Zhang, S.; Qing, Z.; Gao, C.; Zhang, Y.; Zhao, D.; and Sang, N. 2023. MoLo: Motion-augmented Long-short Contrastive Learning for Few-shot Action Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18011–18021.

Wang, X.; Zhang, S.; Qing, Z.; Tang, M.; Zuo, Z.; Gao, C.; Jin, R.; and Sang, N. 2022c. Hybrid Relation Guided Set

Matching for Few-Shot Action Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19948–19957.

Xu, M.; Zhao, C.; Rojas, D. S.; Thabet, A.; and Ghanem, B. 2020. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 10156–10165.

Xu, X.; Wang, J.; Li, X.; and Lu, Y. 2022. Reliable propagation-correction modulation for video object segmentation. In *AAAI*, 1171–1196.

Yang, Z.; Qin, J.; and Huang, D. 2022. ACGNet: Action complement graph network for weakly-supervised temporal action localization. In *AAAI*, 3090–3098.

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, C.-L.; Wu, J.; and Li, Y. 2022. Actionformer: Localizing moments of actions with transformers. In *ECCV*, 492–510.