POTLoc: Pseudo-Label Oriented Transformer for Point-Supervised Temporal Action Localization

Elahe Vahdani[a], Yingli Tian[a]

[a] *The City College, City University of New York, New York, 10031, NY, USA*

Abstract

This paper tackles the challenge of point-supervised temporal action detection, wherein only a single frame is annotated for each action instance in the training set. Most of the current methods, hindered by the sparse nature of annotated points, struggle to effectively represent the continuous structure of actions or the inherent temporal and semantic dependencies within action instances. Consequently, these methods frequently learn merely the most distinctive segments of actions, leading to the creation of incomplete action proposals. This paper proposes POTLoc, a **P**seudo-label **O**riented **T**ransformer for weakly-supervised Action **Loc**alization utilizing only point-level annotation. POTLoc is designed to identify and track continuous action structures via a self-training strategy. The base model begins by generating action proposals solely with point-level supervision. These proposals undergo refinement and regression to enhance the precision of the estimated action boundaries, which subsequently results in the production of 'pseudo-labels' to serve as supplementary supervisory signals. The architecture of the model integrates a transformer with a temporal feature pyramid to capture video snippet dependencies and model actions of varying duration. The pseudo-labels, providing information about the coarse locations and boundaries of actions, assist in guiding the transformer for enhanced learning of action dynamics. POTLoc outperforms the state-of-the-art point-supervised methods on THUMOS'14 and ActivityNet-v1.2 datasets.

Keywords: Temporal action detection, Point-supervised learning, Self-training

1. Introduction

Automated video analysis is attracting substantial attention in the realm of computer vision and multimedia applications, largely due to its potential utility across various fields [1, 2, 3, 4, 5, 6]. A central task in this arena is Temporal Action Localization (TAL) in untrimmed videos, which aims to detect the temporal boundaries of actions and identify their categories [7]. Although recent fully-supervised TAL methods [8, 9, 10] have demonstrated significant progress, they necessitate time-consuming and expensive annotation of temporal boundaries and action labels for each action instance in training videos. To circumvent the requirement for exhaustive annotations, many researchers are gravitating towards the development of weakly-supervised models, which only mandate a minimal set of ground-truth annotations, such as video-level labels. Nonetheless, weakly-supervised models typically lag behind their fully-supervised counterparts in terms of performance, primarily due to limited annotations and the models' constrained capacity to comprehend and learn the structure of actions. To mitigate this performance disparity, the notion of point-level supervision has been introduced [11, 12, 13, 14]. This approach entails annotating a single frame within the temporal window of each action instance in the input video. Even though point-level supervision demands slightly more annotations than weak supervision, it substantially reduces the labeling costs compared to full supervision. Additionally, it imparts vital information about the coarse locations and the overall count of action instances, thereby enriching the model's grasp of action structures.

Due to the sparse nature of annotations in point-level supervision, existing methods frequently fail to effectively model the continuous structure of actions. Prior efforts to augment annotations have involved the generation of pseudo action and background frames, as highlighted in several studies [12, 15, 13, 14]. These pseudo-labeled frames contribute additional supervisory signals to the model, thereby improving its capacity to discern actions from the background. However, in the majority of these approaches, the pseudo-labeled frames are either discontinuous or they cover only fragments of the action intervals. Consequently, they often learn just the most distinctive portions of actions, which ultimately results in the production of incomplete action proposals. To counteract this issue, Lee *et al.*, in [13], developed a framework that employs an action-background contrast method to better understand action completeness, thereby fostering a more comprehensive understanding of action sequences. However, this model still falls short

in adequately representing temporal dependencies within actions.

In this paper, we introduce a point-supervised framework that is designed to capture the continuous structures of actions, even in the face of extremely sparse point-level annotations. Training with only point-level supervision, the base model initially generates noisy action proposals for the training set. These action proposals are subsequently refined and adjusted to generate "pseudo-labels" on the training set using our proposed algorithm. The pseudo-labels represent estimated temporal intervals surrounding the annotated points and are likely to align with action instances. Our pseudo-label generation algorithm is designed to discard the proposals that are potentially redundant, and to adjust those that are either excessively long or overly short. For each annotated point within the training set, we retain only the highest-scoring proposal and adjust its boundaries based on the statistics of the proposals. Importantly, our pseudo-label generation algorithm relies solely on the point-level labels and statistics of the generated proposals. Apart from the given annotated points, no ground-truth labels are used in this step. The generated pseudo-labels act as additional supervisory signals to guide our POTLoc model.

To fully leverage the rich information provided by the pseudo-labels, POTLoc integrates a transformer with a temporal feature pyramid, effectively employing a multi-scale temporal transformer. Training multi-scale transformers for action detection under weak supervision is underexplored due to the scarcity of annotated frames. Our framework shows that we can proficiently train a transformer backbone with sparse point-level annotations. Our transformer utilizes local self-attention, aiding in the modeling of temporal dependencies within video snippets and learning the structure of actions. The temporal feature pyramid facilitates modeling actions of varying duration. The pyramid's lower levels are optimal for detecting shorter actions, while the higher levels, with their larger receptive fields, are suited for modeling longer actions. The pseudo-labels provide information about the coarse location and boundaries of actions, which aids in better guiding our multi-scale temporal transformer to learn action dynamics. Three loss functions are employed to optimize the model to effectively distinguish actions from background and accurately classify different action classes.

We incorporate a sampling strategy during training to select the frames around the annotated points within a radius parameter and inside the boundaries of pseudo-labels, driven by two primary motivations. First, this sampling method selects snippets that are closer to the annotated points (more

indicative of the action) while avoiding farther snippets (action boundaries) that can be ambiguous or contain transitional movements not representative of the action. Our experiments show that the pseudo-labels sampling improves the performance. Second, the sampling helps mitigate the issue of training the model with false positives (background frames incorrectly predicted as actions), which are more likely to exist within the boundaries of pseudo-labels. The main contributions of our work are outlined below.

- We propose an innovative point-supervised framework (POTLoc) to capture the continuous structures of actions, despite relying solely on sparse point-labels.
- We design a novel self-training strategy to generate supplementary supervisory signals (i.e. pseudo-labels) for point-supervised action localization. This is accomplished by refining and adjusting the noisy action proposals, which are predicted by a base point-supervised model on the training set. This procedure is based on analyzing the statistics of the action proposals and their locations in relation to the annotated points.
- Our self-training approach enables the training of a multi-scale transformer backbone with limited supervision. The task of training transformers for action detection under weak supervision was previously underexplored, due to the large number of parameters and the scarcity of annotated frames. The multi-scale temporal transformer, guided by the generated pseudo-labels, learns to model the dependencies of video snippets and actions of varying duration.
- We incorporate a pseudo-labels sampling strategy to mitigate the issue of training the model with false positives and to train the model with more representative snippets.
- POTLoc surpasses the state-of-the-art point-supervised methods on THUMOS'14 and ActivityNet-v1.2 datasets.

2. Related Work

Fully-supervised TAL. Fully-supervised methods are categorized into anchor-based and anchor-free. Anchor-based methods generate dense proposals, distributed across temporal locations [16, 17, 18]. Anchor-free methods employ a bottom-up grouping strategy to generate proposals with precise boundaries and flexible duration [19, 20, 21, 22, 23]. To model actions

with differing duration, temporal feature pyramid was introduced to generate multi-scale temporal features [24, 25, 26, 27]. To model dependencies between video segments, different structures have been utilized, including recurrent neural networks [28, 29], graph convolution networks [30, 31, 23, 32, 33], and transformers [8, 34, 35]. Distinct from these methods that need exhaustive frame-level annotations, our framework utilizes only point-level annotations. Yet, it effectively captures snippet dependencies and models actions of varying duration.

Weakly-supervised TAL. These methods rely on imprecise or coarse labels during the training stage. They often predict attention scores to pinpoint discriminative action regions and eliminate background frames. Attention scores are typically learned through the Multi-Instance Learning (MIL) scheme [36, 37] or via a class-agnostic approach to learn actionness [38, 39, 40, 41]. To model the completeness of actions, several methods have proposed complementary learning approaches aimed at discovering different aspects or parts of actions [42, 43, 44, 45, 46]. Another category of methods relies on an iterative training strategy, which involves generating pseudo-labels from an initial base model to enhance the model's learning capacities [47, 48, 49, 50, 51]. However, these techniques are not capable of generating precise pseudo-labels. Our model generates high-quality pseudo-labels, providing additional guidance to learn the structure of action using slightly more annotations.

Point-supervised TAL. Point-level supervision significantly reduces the cost of annotating action boundaries. Various methods have been proposed to augment annotations: these include the generation of pseudo-actions by expanding annotated frames to their nearby frames [12], or boundary regression based on keyframe prediction [15]. Other strategies include mining pseudo-background frames from unannotated frames [12, 13] or annotating a random frame from a series of consecutive background frames [14]. Lee *et al.* [13] developed an action-background contrast method for to capture action completeness. CRRC-Net [52] proposed a probabilistic pseudo-label mining module to utilize the feature distances from action prototypes to estimate the likelihood of pseudo samples and rectify their corresponding labels for a more reliable classification learning. PCL [53] proposed to generate pseudo labels by estimating the semantic similarity of pair-wise frames in the embedding space. FBI-TAL [54] proposed a pseudo-label search strategy by combining foreground and background labels to exploit the information between them and guide the model. Li *et al.* [55] uses the relationship of the video seg-

ments with their neighbors for pseudo-label generation. Lee *et al.* [56] also uses pseudo-labels for action instance boundary learning.

We propose a self-training framework designed to learn the continuous structure of actions with varying duration using our multi-scale transformer, guided by pseudo-labels. Existing work has utilized pseudo-labels to bridge the gap between classification and temporal localization. The advantages of our framework over previous methods are as follows: 1) The simplicity of the pseudo-label generation module, 2) The ability to capture the completeness of actions using self-training, guided by the estimated pseudo-labels, 3) The capability to model actions of varying duration with point-supervision through the design of a feature pyramid, and 4) The integration of a transformer to capture temporal dependencies under limited supervision.

3. Our Proposed Method

Point-Supervised problem setting. Given an input video, only a single frame is annotated for each action instance, following [12, 15, 13]. Formally, if there are N action instances in the video, the annotation can be denoted by $\{(\epsilon_i, \Lambda_i)\}_{i=1}^{N}$ where ϵ_i is the frame index selected from the temporal interval of $i-$th action instance and Λ_i is the action label. These annotated time-stamps are referred to as "points". Label Λ_i is a binary vector where $\Lambda_i[c]$ is equal to 1 if the label of $i-$th action is c, and 0 otherwise. Video-level labels are given by aggregating the labels of annotated points in each video.

3.1. Point-Supervised Base Model

We employ a base point-supervised model to predict action proposals in the training set. These proposals undergo further refinement, ultimately generating pseudo-labels that serve as augmented supervision for our POTLoc model.

Feature extraction and modeling. The input video is divided into a sequence of snippets, each of which is processed by a pre-trained visual encoder (I3D [57]) for feature extraction. These snippet features are then concatenated to produce a video feature X which is supplied to a shallow temporal convolutional network followed by a sigmoid function. The output results in a class-specific probability signal, $P \in \mathbb{R}^{T \times C+1}$, where $p[t, c]$ represents the probability that snippet t belongs to action class c. T is the number of video snippets and C is the number of action classes. Additionally, $b_t = p[t, C+1]$ is the probability of background at time t. The complement

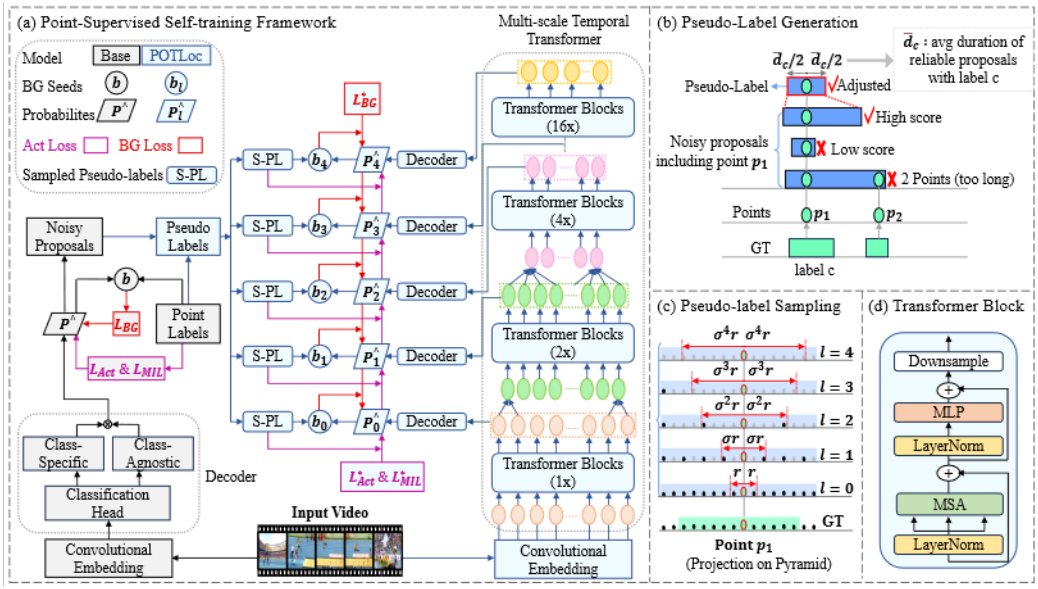


Figure 1: (a) Framework overview. The modules outlined in gray and blue indicate the components of the base and our POTLoc model, respectively. (b) Pseudo-labels are generated from the noisy action proposals predicted by the base model on the training set. The proposals are refined and adjusted based on the point-labels and statistics of the proposals. (c) The pseudo-labels are sampled within a radius around the annotated points at each level l of the pyramid and the block before the pyramid ($l = 0$). This sampling helps to mitigate the addition of excessive noise during training, which could be caused by imprecise estimated action boundaries. (a,d) The multi-scale temporal transformer learns to model temporal dependencies and accommodate actions of varying duration when optimized with our enhanced losses, $\mathcal{L}^*_{\text{MIL}}$, $\mathcal{L}^*_{\text{Act}}$, and $\mathcal{L}^*_{\text{BG}}$ supervised with the pseudo-labels.

of the background score b_t is the class-agnostic score, denoted by a_t. The class-specific and class-agnostic scores are fused to derive the final probability sequence $\hat{P} \in \mathbb{R}^{T\times C+1}$, where $\hat{p}[t, c] = p[t, c] \cdot a_t$ and $\hat{p}[t, C + 1] = b_t$.

Video-level action prediction. We predict a video-level probability vector using the class-specific probability sequence P. For each action class c, we identify the K temporal positions with the highest probability scores. Then, we compute the average score of these positions to represent the video-level probability score for action c, denoted as p^c. Following the MIL scheme [58], a binary cross-entropy loss guides the classification of actions.

$$\mathcal{L}_{\text{MIL}} = -\sum_{c=1}^{C} \Lambda^c \log(p^c) + (1-\Lambda^c)\log(1-p^c). \tag{1}$$

Snippet-level action prediction. Given a video with N annotated points, denoted by $\{(\epsilon_i, \Lambda_i)\}_{i=1}^{N}$, a snippet-level focal loss is employed to optimize the probability signal $\hat{P}$ as follows. γ is the focusing parameter and is set to 2.

$$\begin{aligned}\mathcal{L}_{\text{Act}} = &-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}(1-\hat{p}[\epsilon_i, c])^{\gamma}\Lambda_i[c]\log(\hat{p}[\epsilon_i, c]) \\ &-\hat{p}[\epsilon_i, c]^{\gamma}(1-\Lambda_i[c])\log(1-\hat{p}[\epsilon_i, c]).\end{aligned} \tag{2}$$

Background modeling. To differentiate actions from the background, it is crucial to pinpoint the frames that are likely correlated with the background. However, since there are no explicit annotations for these background frames, we implement a method similar to Lee et al. [13] to generate "background seeds" during training. These seeds are chosen from timestamps that have high background scores surpassing a defined threshold. The predicted background seeds in a given video are denoted by $\{t_j\}_{j=1}^{M}$, and b_{t_j} is the probability of background at time t_j. At these identified time-steps, we suppress the action probabilities and promote the background probabilities by applying the snippet-level focal loss on signal $\hat{P}$.

$$\begin{aligned}\mathcal{L}_{\text{BG}} = &-\frac{1}{M}\sum_{j=1}^{M}\Big[\sum_{c=1}^{C}(\hat{p}[t_j, c])^{\gamma}\log(1-\hat{p}[t_j, c]) \\ &+(1-b_{t_j})^{\gamma}\log b_{t_j}\Big].\end{aligned} \tag{3}$$

Joint training. The total loss for the base model is a weighted combination of the three aforementioned losses, calculated as follows, where $\lambda_\star$ terms balance the losses and are determined through empirical analysis.

$$L_{\text{Total}} = \lambda_{\text{MIL}}\mathcal{L}_{\text{MIL}} + \lambda_{\text{Act}}\mathcal{L}_{\text{Act}} + \lambda_{\text{BG}}\mathcal{L}_{\text{BG}}. \tag{4}$$

Action proposal generation. We set a threshold on the predicted video-level scores to identify the action categories present in the video. Then,

we apply a threshold on the snippet-level action scores $\hat{P}$ for those action categories already predicted. We then merge consecutive candidate segments to form proposals, each of which is assigned a confidence score based on its outer-inner-contrast score [36]. Finally, we use the non-maximum suppression (NMS) technique to eliminate overlapping proposals.

3.2. Pseudo-label Generation from Action Proposals

The base model utilizes point-supervision to predict initial action proposals on the training set. These proposals are redundant and noisy and are unsuitable as pseudo-labels for self-training. In this section, we propose an algorithm to generate pseudo-labels by leveraging the statistics of the proposals and their locations in relation to the annotated points.

Proposal formulation. We define S to be the set of predicted proposals on the training set $\mathcal{V}$. For each video $v \in \mathcal{V}$, S_v denotes the predicted proposal and ϱ_v denotes the points. The predicted start, end, label, and confidence score of the j-th proposal φ_j are denoted by s_j, e_j, Λ_j, and cs_j, respectively. Also, p_i is the i-th point with time ϵ_i and label Λ_i.

$$S_v = \cup_j \{\varphi_j = (s_j, e_j, \Lambda_j, cs_j)\} \ , \ \varrho_v = \cup_i \{p_i = (\epsilon_i, \Lambda_i)\}. \tag{5}$$

Pseudo-label formulation. For a video v, the pseudo-label set is defined as $S_v^* = \{(\epsilon_n, s_n, e_n, \Lambda_n)\}_{n=1}^{N_v}$ where N_v is the number of annotated points in the video and ϵ_n denotes the n-th point. The estimated start, end, and label of the n-th pseudo-label are denoted by s_n, e_n, and Λ_n, respectively. One pseudo-label is generated for each annotated point ϵ_n such that $\epsilon_n \in [s_n, e_n]$. Set S^* is the union of pseudo-labels for all training videos.

$$S^* = \cup_n \{(\epsilon_n, s_n, e_n, \Lambda_n)\}. \tag{6}$$

Pseudo-label generation algorithm. Initially, set S^* only includes the proposals from S that contain precisely one annotated point. These selected proposals are considered more reliable because they are neither excessively long nor too short, avoiding multiple or no points at all. Within S^*, for each action class c, the average duration of proposals with label c, denoted as $\bar{d}_c$, is calculated. We must ensure that each annotated point exists in at least one proposal for complete coverage. Suppose there is a point belonging to action class c with timestamp ϵ_i that is not included in any of the proposals in set S^*. In this case, we search for a list of proposals with label c in the initial set S that include point ϵ_i and select the one with the highest

Algorithm 1 *Pseudo-label Generation*

Input: Proposals S_v, points ϱ_v, for each $v \in \mathcal{V}$
$S_v = \bigcup_j \{\varphi_j = (s_j, e_j, \Lambda_j, cs_j)\}$, $\varrho_v = \bigcup_i \{p_i = (\epsilon_i, \Lambda_i)\}$
Output: Pseudo-labels $S^* = \bigcup_n \{(\epsilon_n, s_n, e_n, \Lambda_n)\}$
Initialization: $S^* = \emptyset$

function $\text{F}(\varphi_j, p_i)$ ▷ (Check if p_i belongs to φ_j)
  if $(\Lambda_i = \Lambda_j) \cap (s_j \leq \epsilon_i \leq e_j)$ **then return** *True*
  else return *False*
  end if
end function
for $\varphi_j \in S_v$ and $v \in \mathcal{V}$: **do**
  if $|\{p_i | p_i \in \varrho_v \text{ s.t. } F(\varphi_j, p_i)\}| = 1$ **then**
    $S^* = S^* \cup \{(\epsilon_i, s_j, e_j, \Lambda_j, cs_j)\}$
  end if
end for
for $c = 1$ to C: **do**
  $\bar{d}_c = \text{mean}(\{(e_j - s_j) | \varphi_j \in S^* \text{ s.t. } (\Lambda_j[c] = 1)\})$
end for
for p_i in ϱ_v and $v \in \mathcal{V}$: **do**
  $\Delta = \bar{d}_c/2 \;\text{ s.t. }\; \Lambda_i[c] = 1$
  if $\{\varphi \in S^* | F(\varphi, p_i)\} = \emptyset$ **then**
    $\tau = \{\varphi \in S | F(\varphi, p_i)\}$
    $k = \text{Argmax}_{cs}(\tau)$ ▷ (proposal with max score)
    $s_k = \max(s_k, \epsilon_i - \Delta)$, $e_k = \min(e_k, \epsilon_i + \Delta)$
    $S^* = S^* \cup \{(\epsilon_i, s_k, e_k, \Lambda_k, cs_k)\}$
  else
    $\tau = \{\varphi \in S^* | F(\varphi, p_i)\}$
    $k = \text{Argmax}_{cs}(\tau)$ ▷ (proposal with max score)
    $S^* = (S^* - \tau) \cup \{(\epsilon_i, s_k, e_k, \Lambda_k, cs_k)\}$
  end if
end for

confidence score. We truncate this proposal within a distance of $\bar{d}_c/2$ from ϵ_i to prevent the new proposal from being too long. All the newly generated proposals from this step are added to set S^*. Finally, we ensure that each annotated point belongs to exactly one proposal by keeping the proposal with the highest confidence score that contains the point and removing the rest.

The confidence scores cs_n were only used for pseudo-label generation and are discarded from S^* at the end. The details of this procedure is summarized in *Algorithm 1*.

3.3. Pseudo-label Oriented Multi-scale Transformer

Fig. 1(a) provides an overview of our framework. The base model utilizes point-supervision to predict initial action proposals. These preliminary action proposals are subsequently used to generate pseudo-labels for the training set based on the point-labels and the statistics of the proposals, as shown in Fig. 1(b). Our POTLoc model employs a multi-scale temporal transformer to capture the temporal dependencies within video snippets and to learn multi-scale temporal action instances, Fig. 1(a,d). POTLoc, supervised by the pseudo-labels, is optimized with three enhanced loss functions, $\mathcal{L}^*_{\text{MIL}}$, $\mathcal{L}^*_{\text{Act}}$, and $\mathcal{L}^*_{\text{BG}}$, to separate actions from background and discriminate actions. Since the pseudo-labels are imprecise estimations of the action boundaries, we sample from the pseudo-labels to mitigate the potential addition of excessive noise during training, Fig. 1(c). The pseudo-labels play a crucial role by equipping the network with detailed information about the approximate locations of actions. This process enhances the effective use of the transformer model and feature pyramid, thereby improving the model's ability to understand action dynamics.

Multi-scale temporal transformer. Given an input video, we extract snippet-level visual features with a pre-trained visual encoder and concatenate them to generate a video feature $X \in \mathbb{R}^T$, where T is the number of snippets. Each snippet feature is embedded using a shallow temporal convolutional network with layer normalization and ReLU, resulting in feature vector $Z^0 \in \mathbb{R}^{T\times D}$. This feature is the input to the transformer network which is employed to model the temporal dependencies between snippets. Feature Z^0 is projected using learnable parameters $W_Q \in \mathbb{R}^{D\times D_q}, W_K \in \mathbb{R}^{D\times D_k}$, and $W_V \in \mathbb{R}^{D\times D_v}$ to extract query, key, and values features, denoted by Q, K, and V, respectively, with $D_q = D_k$. The output of self-attention is $S = \text{Softmax}(QK^T/\sqrt{D_q})V$ where $S \in \mathbb{R}^{T\times D}$. We adapt the local self-attention within a window to reduce the time and memory complexity, following [8]. The transformer network consists of several layers, wherein each layer is composed of multiheaded self-attention (MSA) and MLP blocks, with GELU activation. To model multi-scale features for actions with different duration, we implement down-sampling between transformer blocks using a strided depthwise 1D convolution, resulting in a temporal feature pyramid

$Z = \{Z^1, Z^2, \cdots, Z^L\}$. The l-th transformer block receives the input feature Z^{l-1} and returns the feature Z^l, where $Z^l \in \mathbb{R}^{T_l \times D}$, $T_l = T/\sigma^l$, and σ is the down-sampling ratio. The input to the first transformer block is Z^0. Feature pyramid captures multi-scale temporal information, enabling the model to capture both short-term and long-term temporal dependencies, leading to a more comprehensive representation of action dynamics.

Action decoder. A shallow 1D convolutional network with layer normalization and ReLU is attached to each pyramid level with its parameters shared across all levels. A sigmoid function is attached to each output dimension to predict the probability of actions and background. The output of the l-th level of the feature pyramid is a probability sequence, denoted by $P_l \in \mathbb{R}^{T_l \times C+1}$, where T_l is the temporal dimension on the l-th level. Furthermore, $b_{l,t} = p_l[t, C+1]$ is the probability of background at time t on level l. The class-specific scores are fused with the class-agnostic scores to derive the final probability sequence $\hat{P}_l \in \mathbb{R}^{T_l \times C+1}$.

Pseudo-label sampling. We only consider a narrow interval around the annotated point ϵ_n as a positive instance, as shown in Fig. 1(c). The interval $[\epsilon_n - r, \epsilon_n + r]$ is sampled from pseudo-label interval $[s_n, e_n]$, where r represents the sampling radius and is selected empirically. This sampling procedure mitigates the potential addition of excessive noise during the training, as the interval $[s_n, e_n]$ is merely an approximation of the action boundaries. The projection of the pseudo-label $(\epsilon_n, s_n, e_n, \Lambda_n)$ onto the l-th level of the pyramid becomes $(\epsilon_n/\sigma^l, s_n/\sigma^l, e_n/\sigma^l, \Lambda_n)$ where σ represents the down-sampling ratio. For each level l, we sample an interval with radius $\sigma^l \cdot r$ that is centered around the projected point and located within the projected boundaries. The pseudo-labels at level l are denoted as following.

$$S_l^* = \cup_n \{(\epsilon_n^l, s_n^l, e_n^l, \Lambda_n)\}. \tag{7}$$

Video-level action prediction. The video-level score for class c is defined as the average of $p_l[t_{l,k}^c, c]$ scores where $\{t_{l,k}^c\}_{k=1}^K$ are the top-K positions on level l. The average is calculated over all levels of the pyramid. We utilize the MIL loss (eq. 1) for this extended version and name it $\mathcal{L}_{\text{MIL}}^*$.

Snippet-level action prediction. To simplify the notations, for each level l, we collect all temporal positions of all pseudo-labels into a set, denoted by Φ^l, as follows.

$$\Phi^l = \bigcup_n \{(t, \Lambda_n) |\ t \in [s_n^l, e_n^l] \text{ for } (\epsilon_n^l, s_n^l, e_n^l, \Lambda_n) \in S_l^*\}. \tag{8}$$

We subsequently rename the elements in Φ^l as $\Phi^l = \{(t_m, \Lambda_m)\}_{m=1}^{M_l}$. We extend the snippet-level focal loss (Eq. 2) to all temporal positions of all pseudo-labels to optimize the learning of the probability signal $\hat{P}_l$ for each level l of the pyramid. M is the total number of positive instances.

$$\begin{aligned}\mathcal{L}^*_{\text{Act}} = -\frac{1}{M}\sum_{l=1}^{L}\sum_{m=1}^{M_l}\sum_{c=1}^{C}&(1-\hat{p}_l[t_m,c])^{\gamma}\Lambda_m[c]\log(\hat{p}_l[t_m,c]) \\ &-\hat{p}_l[t_m,c]^{\gamma}(1-\Lambda_m[c])\log(1-\hat{p}_l[t_m,c]).\end{aligned} \tag{9}$$

Background modeling. To distinguish actions from the background, similar to the base model, we select the temporal positions not belonging to any of the pseudo-labels and possessing a background probability exceeding a certain threshold on each level l of the pyramid. The background loss presented in Eq. 3 is extended to all pyramid levels to optimize the probability signal $\hat{P}_l$, and is denoted by $\mathcal{L}^*_{\text{BG}}$.

Joint training. Our POTLoc model is trained using a combination of the three enhanced losses with $\lambda_\star$ weighting parameters that are determined through empirical analysis.

$$L_{\text{Total}} = \lambda_{\text{MIL}}\mathcal{L}^*_{\text{MIL}} + \lambda_{\text{Act}}\mathcal{L}^*_{\text{Act}} + \lambda_{\text{BG}}\mathcal{L}^*_{\text{BG}}. \tag{10}$$

Inference. The action categories are identified using the video-level scores. The action proposals are predicted from all pyramid levels by applying thresholds to the snippet-level action scores $\hat{P}_l$ for each level l for the predicted classes. The strategy used is similar to the inference of the base model.

4. Experiments

4.1. Experimental Setting

Datasets. THUMOS14 consists of untrimmed videos spanning 20 distinct categories. Following previous methods [13, 47], we utilized the validation set for training, and the testing set for evaluation. ActivityNet-v1.2 is a large-scale dataset encompassing 100 complex daily activities. We follow the convention of using the training set to train our model, and the validation set for evaluation [13, 47].

Evaluation metric. The Mean Average Precision (mAP) under different Intersection over Union (IoU) thresholds is utilized as the evaluation metric, wherein the Average Precision (AP) is computed for each action class.

Implementation details. For feature extraction, the two-stream I3D model [57] is utilized on both datasets. Segments consisting of 16 consecutive frames are fed as input to the visual encoder, employing a sliding window approach with a stride of 16 on both THUMOS14 and ActivityNet-v1.2. Both base and POTLoc models are optimized by Adam [59] with the learning rate of 10^{-4} for 50 epochs. In the base model, the original number of feature segments is used without sampling. However, in the main model, the input length is set to 768 for THUMOS14 and to 192 for ActivityNet-v1.2, using random sampling and linear interpolation. A window of 12 and 7 is used for local self-attention on THUMOS14 and ActivityNet-v1.2, respectively. In POTLoc model, the number of pyramid levels is set to $l = 2$ and the sampling radius is set to $r = 2$. The parameter r is defined on the feature grid, representing the distance in terms of the number of features. The batch size is set to 4 on THUMOS14, and to 64 on ActivityNet-v1.2. At inference time, the full sequence is fed into the model without sampling. The source code will be released once the paper is accepted.

Computational Complexity: The number of parameters and FLOPs are 17.9B and 12.6M for the base model and 37.5B and 24.4M for multi-scale transformer.

4.2. Comparison with State-of-the-art Methods

Table 1 presents a comprehensive comparison with the state-of-the-art methods on THUMOS'14 and ActivityNet-v1.2.

Results on THUMOS'14. Our model achieves state-of-the-art performance among weakly-supervised and point-supervised methods in terms of average mAP. Additionally, our model demonstrates remarkable results of an 6% average mAP increase compared to weakly-supervised methods, despite only using slightly more annotations.

Results on ActivityNet-v1.2. Our model outperforms all the state-of-the-art weakly and point-supervised methods in terms of mAP, consistently across all the IoU thresholds. We note that the performance gains over weakly-supervised methods on ActivityNet-v1.2 are smaller compared to those on the THUMOS'14 dataset. This is primarily because the average number of action instances per video in THUMOS'14 is higher than that in ActivityNet-v1.2 (15.5 vs. 1.5). Consequently, on THUMOS'14, the model

Group	Method	THUMOS'14						ActivityNet-v1.2			
		mAP@IoU (%)					mAP-AVG	mAP@IoU (%)			mAP-AVG
		0.3	0.4	0.5	0.6	0.7	(0.1:0.7)	0.5	0.75	0.95	(0.5:0.95)
WS	ASL[60]	51.8	-	31.1	-	11.4	40.3	40.2	-	-	25.8
	CoLA [61]	51.5	41.9	32.2	22.0	13.1	40.9	42.7	25.7	5.8	26.1
	AUMN [39]	54.9	44.4	33.3	20.5	9.0	41.5	42.0	25.0	5.6	25.5
	FTCL [62]	55.2	45.2	35.6	23.7	12.2	43.6	-	-	-	-
	UGCT [50]	55.5	46.5	35.9	23.8	11.4	43.6	41.8	25.3	5.9	25.8
	CO2-Net [38]	58.2	47.1	35.9	23.0	12.8	-	43.3	26.3	5.2	26.4
	D2-Net [40]	52.3	43.4	36.0	-	-	-	42.3	25.5	5.8	26.0
	ASM-Loc[47]	57.1	46.8	36.6	25.2	13.4	45.1	-	-	-	-
	RSKP[63]	55.8	47.5	38.2	25.4	12.5	45.1	-	-	-	-
	TS[64]	60.0	47.9	37.1	24.4	12.7	46.2	-	-	-	-
	DELU[65]	56.5	47.7	40.5	27.2	15.3	46.4	44.2	26.7	5.4	26.9
	P-MIL [66]	58.9	49.0	40.0	27.1	15.1	47.0	44.2	26.1	5.3	26.5
	Zhou *et al.* [67]	60.7	51.8	42.7	26.2	13.1	48.3	-	-	-	-
	PivoTAL [68]	61.7	52.1	42.8	30.6	16.7	49.6	-	-	-	-
PS	SF-Net [12]	52.8	42.2	30.5	20.6	12.0	41.2	37.8	-	-	22.8
	DCM [69]	58.1	46.4	34.5	21.8	11.9	44.3	-	-	-	-
	PTAL [15]	58.2	47.1	35.9	23.0	12.8	-	-	-	-	-
	BackTAL [14]	54.4	45.5	36.3	26.2	14.8	-	41.5	27.3	4.7	27.0
	PCL [53]	63.3	55.9	44.4	-	-	-	-	-	-	-
	Lee *et al.* [13]	64.6	56.5	45.3	34.5	21.8	52.8	44.0	26.0	5.9	26.8
	CRRC-Net [52]	67.1	57.9	46.6	33.7	19.8	53.8	-	-	-	-
	Lee *et al.* [56]	66.8	57.8	47.1	34.8	21.1	-	44.6	26.7	6.1	27.2
	FBI-TAL [54]	66.7	58.3	48.3	36.3	21.9	54.6	-	-	-	-
	Li *et al.* [55]	66.6	59.4	48.6	36.7	**22.7**	55.1	43.4	31.3	5.4	27.5
	POTLoc	**68.8**	**59.5**	**50.1**	**37.1**	21.2	**55.7**	**45.1**	**27.6**	**6.8**	**28.0**

Table 1: Comparison with weakly-supervised (WS) and point-supervised (PS) methods on THUMOS'14 and ActivityNet-v1.2. The results are reported in terms of mAP (%) at different tIoU thresholds. The bold numbers show the best results.

can learn to distinguish actions from the background with the assistance of inferred background seeds situated between consecutive action points. This is more challenging on ActivityNet-v1.2 due to the sparse nature of action instances.

4.3. Ablation Studies

We conduct ablation studies on THUMOS'14 to analyze the impact of each component of the proposed model.

Pseudo-label generation. Table 2 demonstrates the quality of the generated pseudo-labels. This table reports the performance on the train set (validation set) of THUMOS'14. α represents the ratio of the number of generated proposals to the ground-truth instances. The noisy proposals are predicted by the base model without refinement. The refinement significantly

Step	$\alpha = \frac{\#\text{Proposals}}{\#\text{GT}}$	mAP@0.5 (%)	mAP-AVG (%) (0.1:0.7)
Noisy Proposals	~ 12	57.0	63.5
Pseudo-labels	~ 1	62.7	69.5

Table 2: Analysis of pseudo-labels on the **validation set** of THUMOS'14.

Backbone, Losses	Supervision	SL	mAP(%)
Multi-scale Transformer, Enhanced Losses: $\{\mathcal{L}^*_{\text{MIL}}, \mathcal{L}^*_{\text{Act}}, \mathcal{L}^*_{\text{BG}}\}$	Ground-truth	✓	56.0
		✗	52.1
	Pseudo-labels **(POTLoc)**	✓	**55.7**
		✗	51.0
	Noisy Proposals	✓	26.8
		✗	38.3
	Points	✗	50.4
Temporal Convolutions, Base Losses: $\{\mathcal{L}_{\text{MIL}}, \mathcal{L}_{\text{Act}}, \mathcal{L}_{\text{BG}}\}$	Ground-truth	✓	50.1
		✗	44.8
	Pseudo-labels	✓	49.8
		✗	46.9
	Noisy Proposals	✓	28.7
		✗	38.2
	Points	✗	47.4

Table 3: Impact of the main components of our framework on THUMOS'14, measured in terms of average mAP. SL denotes sampling, with the radius set to 2. The bold number represents the performance of our full POTLoc model.

removes redundant proposals and improves the alignments with ground-truth intervals. The pseudo-labels provide exactly one interval around each annotated point and $\alpha = 1$. Furthermore, the performance of the pseudo-labels is 6% average mAP higher than the noisy proposals which highlights the effectiveness of the proposed pseudo-label generation method.

Impact of pseudo-labels. Table 3 highlights the crucial role of pseudo-label generation. It is worth noting that the use of noisy proposals results in poor performance, underperforming the models supervised with single points. This is because noisy proposals provide a highly inaccurate estimation of action boundaries. Moreover, many of these proposals may be redundant and overlapping. This highlights the importance of proposal refinement in our pseudo-label generation. To further assess the quality of the pseudo-

Radius	mAP@IoU (%)					mAP-AVG (%)
	0.3	0.4	0.5	0.6	0.7	(0.1:0.7)
$r = 2$	**68.8**	**59.5**	**50.1**	**37.1**	**21.2**	**55.7**
$r = 4$	65.4	56.8	46.0	34.0	18.7	53.0
$r = \infty$	63.0	53.8	43.5	32.3	18.4	51.0

Table 4: Impact of the pseudo-label sampling radius r in POTLoc model on THUMOS'14.

labels, we conduct experiments using ground-truth labels. We observe that the performance of the model supervised by pseudo-labels is comparable with that of the fully-supervised model. This can be attributed to our model not depending on information about the precise location of action boundaries, which could otherwise be employed in a regression loss.

Label sampling. The impact of sampling across different supervision levels is demonstrated in Table 3. Sampling consistently improves the performance for both pseudo-labels and ground-truth labels. When pseudo-labels are utilized, sampling mitigates the noise introduced by imprecise action boundaries. Moreover, when using ground-truth labels, sampling encourages higher scores around action centers, encouraging the model to learn meaningful and representative action snippets. In other words, sampling selects snippets that are closer to the action centers (often more indicative of the action) while avoiding boundary snippets that can be ambiguous or contain transitional movements not representative of the action. Therefore, sampling improves the performance even in the case of training with ground-truth boundaries. However, sampling does not enhance performance when the model is supervised with noisy proposals. This is primarily because the center of the noisy proposals may not necessarily be close to the center of the action instances. In this scenario, sampling may inadvertently lead to a focus on a random video snippet such as background. Moreover, Table 4 demonstrates the importance of the sampling strategy with different sampling radius r, which $r = \infty$ indicates *no sampling*.

The backbone architecture. Table 3 illustrates that the multi-scale transformer when trained with enhanced losses, achieves significantly better results compared to the base model. The latter only consists of convolutional layers and is trained with base losses. The performance enhancement is consistent across ground-truth, pseudo-labels, and points supervision. However, for noisy proposals, the results of different models are comparable. Table 5 demonstrates the impact of the number of pyramid levels, denoted by l, in

Levels	$l=0$	$l=1$	$l=2$	$l=3$	$l=4$
mAP@0.7	18.8	18.4	**21.2**	20.7	19.25
mAP-AVG	51.2	54.7	**55.7**	55.7	54.1

Table 5: Impact of the number of pyramid levels (denoted by l) on THUMOS'14. The backbone is POTLOC's multi-scale transformer supervised with pseudo-labels.

Loss Parameters			mAP-AVG (%)
λ_{MIL}	λ_{Act}	λ_{BG}	(0.1:0.7)
1	1	1	55.2
0.5	1	1	54.2
1	0.5	1	**55.7**
1	1	0.5	52.7

Table 6: Impact of the loss functions in POTLoc on THUMOS'14.

POTLoc. The model denoted by $l=0$ incorporates transformer blocks without a feature pyramid, leading to the lowest performance. The model with $l=2$ achieves the highest performance at an IoU of 0.7 reflecting generation of complete action proposals with the assistance of the feature pyramid. Our findings suggest that adding more pyramid levels ($l \geq 3$) does not improve the performance further.

Impact of the loss functions. As mentioned earlier, our POTLoc model is trained using a combination of three loss functions $\mathcal{L}^*_{\text{MIL}}$, $\mathcal{L}^*_{\text{Act}}$, and $\mathcal{L}^*_{\text{BG}}$ (eq. 9). Table 6 reports the impact of the $\lambda_\star$ weighting parameters. The highest average mAP is achieved when $\lambda_{\text{MIL}}=1$, $\lambda_{\text{Act}}=0.5$ and $\lambda_{\text{BG}}=1$.

Distribution of annotated points. In the point-supervision setting, only a single frame per action instance is annotated in the training set. SF-Net [12] proposed to simulate point annotations by sampling a single frame for each action instance. The Uniform distribution method randomly selects a frame within the action boundaries of each action, while the Gaussian distribution method does so with respect to a given mean and standard deviation. Typically, the Gaussian distribution is more likely to sample frames closer to the central timestamps of actions, thereby increasing the chances of choosing a more discriminative snippet. In contrast, the Uniform distribution can sample frames from any part of the action, without this central bias. Table 7 demonstrates that POTLoc attains state-of-the-art results with both Uniform and Gaussian point-level distributions on THUMOS'14, indicating its

Distribution	Method	mAP@IoU (%)			mAP-AVG (%)
		0.3	0.5	0.7	(0.1:0.7)
Gaussian	**POTLoc**	**68.8**	**50.1**	**21.2**	**55.7**
	LACP [13]	64.6	45.3	**21.8**	52.8
Uniform	**POTLoc**	**64.1**	**43.5**	17.7	**51.3**
	LACP [13]	60.4	42.6	**20.2**	49.3

Table 7: Performance comparison with uniform and Gaussian point-level distributions on THUMOS'14.

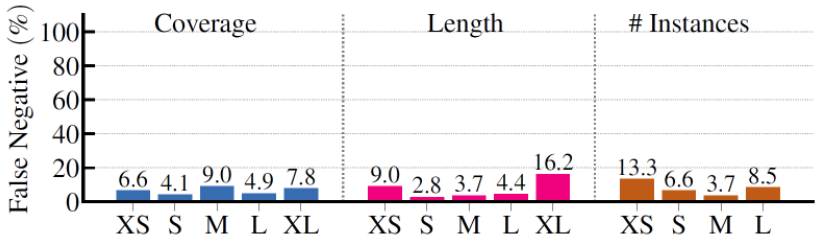
robustness. However, it is observed that the POTLoc's performance is lower with the Uniform distribution as compared to the Gaussian distribution. We conjecture this may be attributed to the Uniform distribution's tendency to select less discriminative snippets for point annotation, which can occur anywhere within the action's extent, such as at the boundaries. Bridging the performance gap between models trained with different sampling distributions of annotated points (Gaussian and Uniform) can be considered for future work.

4.4. Temporal Action Detection Error Analysis

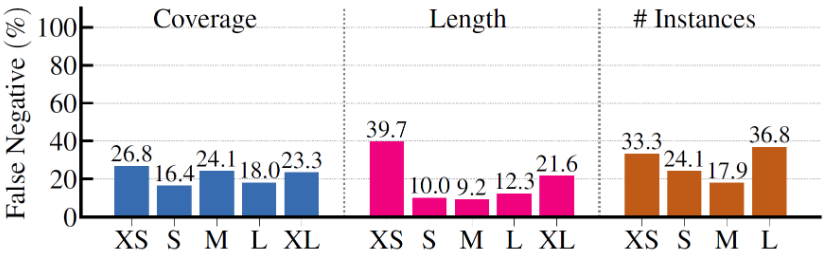
DETAD [70] is empolyed for analyzing false negatives (Fig. 2) and false positives (Fig. 3) of POTLoc in comparison with the base model and a fully-supervised method (ActionFormer[8]).

False Negative Analysis. Fig. 2 illustrates the false negative (FN) profiling across various coverages, lengths, and number of instances. Part (b) of Fig. 2 displays the FN profiling of POTLoc. The figure reveals that higher false negative rates are associated with action instances characterized by: (1) extremely short (Coverage (XS)) or extremely long (Coverage (XL)) durations relative to the video length , (2) actions of very short or very long lengths (Length (XS) or Length (XL)), and (3) videos containing very small (#Instances (XS)) or large number of action instances (#Instances (L)). Furthermore, Fig. 2 demonstrates that POTLoc (part b) reduces the false negative (FN) rate compared to the base model (part c) in most cases. FN profiling of ActionFormer[8] is provided (part a) which has much lower false negative rate compared with POTLoc because of access to the annotation of action boundaries.

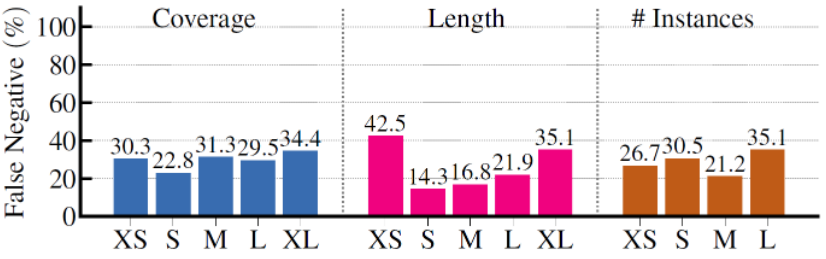
False Positive Analysis. Fig. 3 presents a detailed categorization of false positive errors and summarizes their distribution. In comparison with



(a) ActionFormer [8] (Fully-supervised).



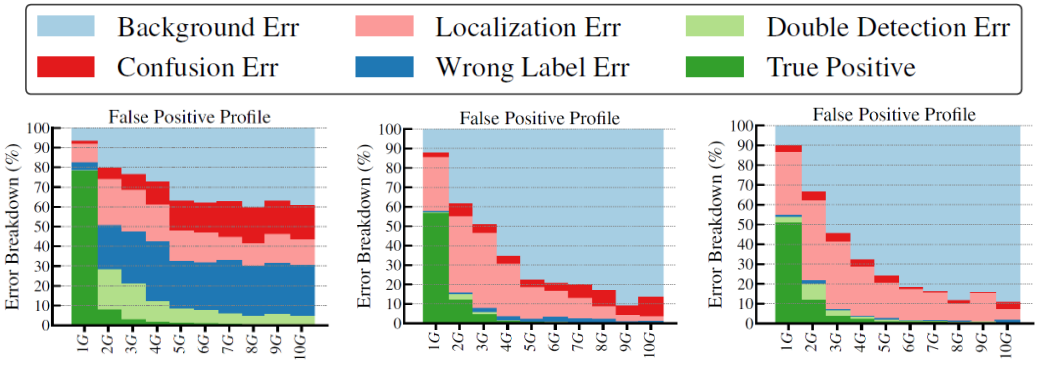
(b) Our POTLoc model (Point-supervised).



(c) Our base model (Point-supervised).

Figure 2: False negative profiling of ActionFormer [8] (fully-supervised), POTLoc (point-supervised) and the base model (point-supervised) on THUMOS14 using DETAD [70].

Actionformer (part a), the majority of false positive errors in POTLoc (part b) stem from background errors. This occurs because POTLoc lacks access to precise action boundaries. Therefore, background snippets close to action boundaries may be erroneously detected as actions, resulting in false positives. Moreover, the false positive profiling of POTLoc (part b) is compared against the base model (part c). POTLoc detects more true positive instances and exhibits fewer localization and confusion errors which confirms the effectiveness of POTLoc compared to the base model.



(a) ActionFormer[8](Supervised) (b) POTLoc (Point-supervised) (c) Base model (Point-supervised)

Figure 3: False positive (FP) profiling of ActionFormer [8] (fully-supervised), POTLoc (point-supervised) and base model (point-supervised) on THUMOS14 using DETAD [70].

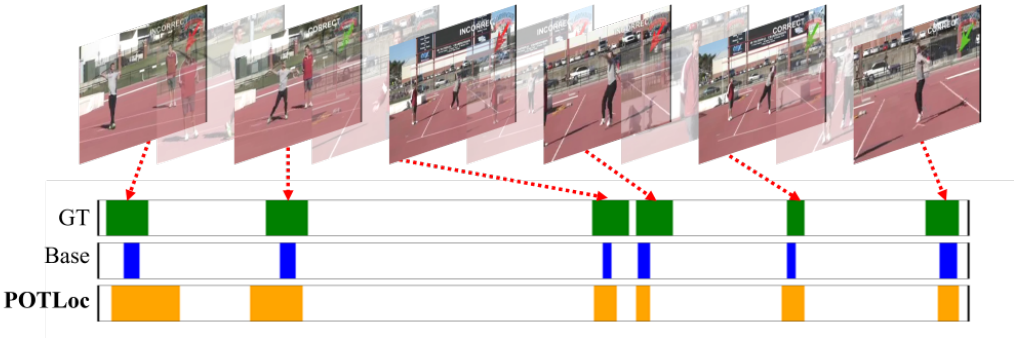
4.5. Qualitative Results

Fig. 4 presents the qualitative results of our model in comparison with the base model. POTLoc addresses various types of errors in the base model such as incompleteness and misalignment. In some cases, POTLoc successfully detects complete action proposals, whereas the base model tends to detect fragmented and disconnected segments of action instances. However, as a limitation of POTLoc, in some cases the predicted proposals are over-completed (expanded beyond the action boundaries).

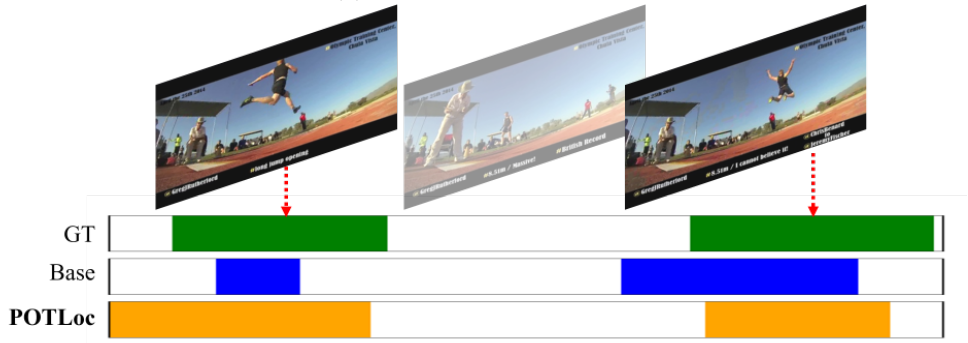
5. Conclusion

We have proposed a novel point-supervised framework, POTLoc, that employs a self-training scheme to effectively learn action dynamics. A unique strategy is formulated for pseudo-label generation, which refines action proposals generated from the base model, thus offering supplemental supervisory signals. The effectiveness of the proposed approach for generating and sampling pseudo-labels is confirmed through our experiments. We further elucidated how the transformer and the feature pyramid network utilize the guidance from pseudo-labels to accurately model continuous action structures and handle actions of various durations. POTLoc outperforms the state-of-the-art methods on THUMOS'14 dataset and ActivityNet-v1.2.

(a) Action "Billiards".



(b) Action "Javelin Throw".



(c) Action "Long Jump".

Figure 4: Qualitative results on THUMOS'14. The ground-truth instances are highlighted in green. The detection results are displayed from: (1) the base model supervised with point-level annotations (blue), and (2) our POTLoc framework (orange). Transparent frames represent background frames.

6. Acknowledgment

This material is based upon work supported by the National Science Foundation under award number 2041307.

References

[1] C. He, J. Shao, J. Sun, An anomaly-introduced learning method for abnormal event detection, Multimedia Tools and Applications 77 (22) (2018) 29573–29588.

[2] A. Cioppa, A. Deliege, S. Giancola, B. Ghanem, M. V. Droogenbroeck, R. Gade, T. B. Moeslund, A context-aware loss function for action spotting in soccer videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13126–13136.

[3] S. Giancola, M. Amine, T. Dghaily, B. Ghanem, Soccernet: A scalable dataset for action spotting in soccer videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1711–1721.

[4] A. Rasouli, J. K. Tsotsos, Autonomous vehicles that interact with pedestrians: A survey of theory and practice, IEEE Transactions on Intelligent Transportation Systems 21 (3) (2019) 900–918.

[5] K. Mahadevan, E. Sanoubari, S. Somanath, J. E. Young, E. Sharlin, Av-pedestrian interaction design using a pedestrian mixed traffic simulator, in: Proceedings of the 2019 on Designing Interactive Systems Conference, 2019, pp. 475–486.

[6] Y. Yao, X. Wang, M. Xu, Z. Pu, E. Atkins, D. Crandall, When, where, and what? a new dataset for anomaly detection in driving videos, arXiv preprint arXiv:2004.03044 (2020).

[7] E. Vahdani, Y. Tian, Deep learning-based action detection in untrimmed videos: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (4) (2023) 4302–4320.

[8] C.-L. Zhang, J. Wu, Y. Li, Actionformer: Localizing moments of actions with transformers, in: Computer Vision–ECCV 2022: 17th European

Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV, Springer, 2022, pp. 492–510.

[9] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, Y. Qiao, Videomae v2: Scaling video masked autoencoders with dual masking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14549–14560.

[10] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Li, D. Tao, Tridet: Temporal action detection with relative boundary modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 18857–18866.

[11] D. Moltisanti, S. Fidler, D. Damen, Action recognition from single timestamp supervision in untrimmed videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9915–9924.

[12] F. Ma, L. Zhu, Y. Yang, S. Zha, G. Kundu, M. Feiszli, Z. Shou, Sf-net: Single-frame supervision for temporal action localization, in: European conference on computer vision, Springer, 2020, pp. 420–437.

[13] P. Lee, H. Byun, Learning action completeness from points for weakly-supervised temporal action localization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13648–13657.

[14] L. Yang, J. Han, T. Zhao, T. Lin, D. Zhang, J. Chen, Background-click supervision for temporal action localization, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[15] C. Ju, P. Zhao, Y. Zhang, Y. Wang, Q. Tian, Point-level temporal action localization: Bridging fully-supervised proposals to weakly-supervised losses, arXiv preprint arXiv:2012.08236 (2020).

[16] J. Gao, Z. Yang, K. Chen, C. Sun, R. Nevatia, Turn tap: Temporal unit regression network for temporal action proposals, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3628–3636.

[17] J. Gao, Z. Yang, R. Nevatia, Cascaded boundary regression for temporal action detection, arXiv preprint arXiv:1705.01180 (2017).

[18] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, R. Sukthankar, Rethinking the faster r-cnn architecture for temporal action localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1130–1139.

[19] T. Lin, X. Zhao, H. Su, C. Wang, M. Yang, Bsn: Boundary sensitive network for temporal action proposal generation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.

[20] T. Lin, X. Liu, X. Li, E. Ding, S. Wen, Bmn: Boundary-matching network for temporal action proposal generation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3889–3898.

[21] C. Lin, J. Li, Y. Wang, Y. Tai, D. Luo, Z. Cui, C. Wang, J. Li, F. Huang, R. Ji, Fast learning of temporal action proposal via dense boundary generator., in: AAAI, 2020, pp. 11499–11506.

[22] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Fu, Learning salient boundary feature for anchor-free temporal action localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3320–3329.

[23] Y. Bai, Y. Wang, Y. Tong, Y. Yang, Q. Liu, J. Liu, Boundary content graph neural network for temporal action proposal generation, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16, Springer, 2020, pp. 121–137.

[24] T. Lin, X. Zhao, Z. Shou, Single shot temporal action detection, in: Proceedings of the 25th ACM international conference on Multimedia, 2017, pp. 988–996.

[25] D. Zhang, X. Dai, X. Wang, Y.-F. Wang, S3d: single shot multi-span detector via fully 3d convolutional networks, arXiv preprint arXiv:1807.08069 (2018).

[26] Y. Liu, L. Ma, Y. Zhang, W. Liu, S.-F. Chang, Multi-granularity generator for temporal action proposal, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3604–3613.

[27] Q. Liu, Z. Wang, Progressive boundary refinement network for temporal action detection, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 34, 2020, pp. 11612–11619.

[28] S. Buch, V. Escorcia, C. Shen, B. Ghanem, J. Carlos Niebles, Sst: Single-stream temporal action proposals, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 2911–2920.

[29] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, J. C. Niebles, End-to-end, single-stream temporal action detection in untrimmed videos (2019).

[30] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, C. Gan, Graph convolutional networks for temporal action localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 7094–7103.

[31] J. Li, X. Liu, Z. Zong, W. Zhao, M. Zhang, J. Song, Graph attention based proposal 3d convnets for action detection., in: AAAI, 2020, pp. 4626–4633.

[32] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, B. Ghanem, G-tad: Sub-graph localization for temporal action detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10156–10165.

[33] C. Zhao, A. K. Thabet, B. Ghanem, Video self-stitching graph network for temporal action localization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13658–13667.

[34] M. Nawhal, G. Mori, Activity graph transformer for temporal action localization, arXiv preprint arXiv:2101.08540 (2021).

[35] S. Chang, P. Wang, F. Wang, H. Li, Z. Shou, Augmented transformer with adaptive graph for temporal action proposal generation, in: Proceedings of the 3rd International Workshop on Human-Centric Multimedia Analysis, 2022, pp. 41–50.

[36] P. Lee, Y. Uh, H. Byun, Background suppression network for weakly-supervised temporal action localization., in: AAAI, 2020, pp. 11320–11327.

[37] S. Narayan, H. Cholakkal, F. S. Khan, L. Shao, 3c-net: Category count and center loss for weakly-supervised action localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8679–8687.

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

[39] W. Luo, T. Zhang, W. Yang, J. Liu, T. Mei, F. Wu, Y. Zhang, Action unit memory network for weakly supervised temporal action localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9969–9979.

[40] S. Narayan, H. Cholakkal, M. Hayat, F. S. Khan, M.-H. Yang, L. Shao, D2-net: Weakly-supervised action localization via discriminative embeddings and denoised activations, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13608–13617.

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

[42] K. K. Singh, Y. J. Lee, Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization, in: 2017 IEEE international conference on computer vision (ICCV), IEEE, 2017, pp. 3544–3553.

[43] J.-X. Zhong, N. Li, W. Kong, T. Zhang, T. H. Li, G. Li, Step-by-step erasion, one-by-one collection: a weakly supervised temporal action detector, in: Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 35–44.

[44] K. Min, J. J. Corso, Adversarial background-aware loss for weakly-supervised temporal activity localization, in: Computer Vision–ECCV

2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, Springer, 2020, pp. 283–299.

[45] R. Zeng, C. Gan, P. Chen, W. Huang, Q. Wu, M. Tan, Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization, IEEE Transactions on Image Processing 28 (12) (2019) 5797–5808.

[46] D. Liu, T. Jiang, Y. Wang, Completeness modeling and context separation for weakly supervised temporal action localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1298–1307.

[47] B. He, X. Yang, L. Kang, Z. Cheng, X. Zhou, A. Shrivastava, Asm-loc: Action-aware segment modeling for weakly-supervised temporal action localization, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 13925–13935.

[48] Z. Luo, D. Guillory, B. Shi, W. Ke, F. Wan, T. Darrell, H. Xu, Weakly-supervised action localization with expectation-maximization multi-instance learning, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16, Springer, 2020, pp. 729–745.

[49] A. Pardo, H. Alwassel, F. Caba, A. Thabet, B. Ghanem, Refineloc: Iterative refinement for weakly-supervised action localization, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 3319–3328.

[50] W. Yang, T. Zhang, X. Yu, T. Qi, Y. Zhang, F. Wu, Uncertainty guided collaborative training for weakly supervised temporal action detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 53–63.

[51] Y. Zhai, L. Wang, W. Tang, Q. Zhang, J. Yuan, G. Hua, Two-stream consensus network for weakly-supervised temporal action localization, in: European conference on computer vision, Springer, 2020, pp. 37–54.

[52] J. Fu, J. Gao, C. Xu, Compact representation and reliable classification learning for point-level weakly-supervised action localization, IEEE Transactions on Image Processing 31 (2022) 7363–7377.

[53] P. Li, J. Cao, X. Ye, Prototype contrastive learning for point-supervised temporal action detection, Expert Systems with Applications 213 (2023) 118965.

[54] Y. Dong, G. Li, F. Wang, W. Wen, X. Xu, L. Feng, Fbi-tal: Foreground-background integration for single-frame supervised temporal action localization, in: 2023 18th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), IEEE, 2023, pp. 394–401.

[55] G. Li, D. Cheng, N. Wang, J. Li, X. Gao, Neighbor-guided pseudo-label generation and refinement for single-frame supervised temporal action localization, IEEE Transactions on Image Processing (2024).

[56] S. Lee, J. Lim, J. Moon, C. Jung, An improved point-level supervision method for temporal action localization, IEEE Access (2023).

[57] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.

[58] T. G. Dietterich, R. H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artificial intelligence 89 (1-2) (1997) 31–71.

[59] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[60] J. Ma, S. K. Gorti, M. Volkovs, G. Yu, Weakly supervised action selection learning in video, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7587–7596.

[61] C. Zhang, M. Cao, D. Yang, J. Chen, Y. Zou, Cola: Weakly-supervised temporal action localization with snippet contrastive learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16010–16019.

[62] J. Gao, M. Chen, C. Xu, Fine-grained temporal contrastive learning for weakly-supervised temporal action localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19999–20009.

[63] L. Huang, L. Wang, H. Li, Weakly supervised temporal action localization via representative snippet knowledge propagation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3272–3281.

[64] Y. Wang, Y. Li, H. Wang, Two-stream networks for weakly-supervised temporal action localization with semantic-aware mechanisms, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 18878–18887.

[65] M. Chen, J. Gao, S. Yang, C. Xu, Dual-evidential learning for weakly-supervised temporal action localization, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV, Springer, 2022, pp. 192–208.

[66] H. Ren, W. Yang, T. Zhang, Y. Zhang, Proposal-based multiple instance learning for weakly-supervised temporal action localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2394–2404.

[67] J. Zhou, L. Huang, L. Wang, S. Liu, H. Li, Improving weakly supervised temporal action localization by bridging train-test gap in pseudo labels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 23003–23012.

[68] M. N. Rizve, G. Mittal, Y. Yu, M. Hall, S. Sajeev, M. Shah, M. Chen, Pivotal: Prior-driven supervision for weakly-supervised temporal action localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 22992–23002.

[69] C. Ju, P. Zhao, S. Chen, Y. Zhang, Y. Wang, Q. Tian, Divide and conquer for single-frame temporal action localization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13455–13464.

[70] H. Alwassel, F. Caba Heilbron, V. Escorcia, B. Ghanem, Diagnosing error in temporal action detectors, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 256–272.