International Journal of Computer Vision (2023) 131:302–323
https://doi.org/10.1007/s11263-022-01702-9

AOE-Net: Entities Interactions Modeling with Adaptive Attention Mechanism for Temporal Action Proposals Generation

Khoa Vo[1] · Sang Truong[1] · Kashu Yamazaki[1] · Bhiksha Raj[2] · Minh-Triet Tran[3,4] · Ngan Le[1]

Received: 21 January 2022 / Accepted: 30 September 2022 / Published online: 28 October 2022

Abstract

Temporal action proposal generation (TAPG) is a challenging task, which requires localizing action intervals in an untrimmed video. Intuitively, we as humans, perceive an action through the interactions between actors, relevant objects, and the surrounding environment. Despite the significant progress of TAPG, a vast majority of existing methods ignore the aforementioned principle of the human perceiving process by applying a backbone network into a given video as a black-box. In this paper, we propose to model these interactions with a multi-modal representation network, namely, *Actors-Objects-Environment Interaction Network (AOE-Net)*. Our AOE-Net consists of two modules, i.e., perception-based multi-modal representation (PMR) and boundary-matching module (BMM). Additionally, we introduce *adaptive attention mechanism (AAM)* in PMR to focus only on main actors (or relevant objects) and model the relationships among them. PMR module represents each video snippet by a visual-linguistic feature, in which main actors and surrounding environment are represented by visual information, whereas relevant objects are depicted by linguistic features through an image-text model. BMM module processes the sequence of visual-linguistic features as its input and generates action proposals. Comprehensive experiments and extensive ablation studies on ActivityNet−1.3 and THUMOS-14 datasets show that our proposed AOE-Net outperforms previous state-of-the-art methods with remarkable performance and generalization for both TAPG and temporal action detection. To prove the robustness and effectiveness of AOE-Net, we further conduct an ablation study on egocentric videos, i.e. EPIC-KITCHENS 100 dataset. Our source code is publicly available at https://github.com/UARK-AICV/AOE-Net.

Keywords Temporal action proposal · Temporal action detection · Human perceiving process · Attention mechanism · Human · Objects · Environment · Interaction · Video understanding

1 Introduction

Given an untrimmed video, TAPG targets localizing temporal segments with specific starting and ending timestamps for each action or activity appearing in the video. TAPG has emerged as one of the most important problems in video analysis and understanding (Shou et al., 2016; Gao et al., 2017, 2018a, b). More specifically, TAPG is a key module for other downstream tasks including temporal action detection (TAD) (Fabian Caba Heilbron and Niebles, 2015; Jiang et al., 2014), video captioning (Krishna et al., 2017),

Communicated by Liwei Wang.

Sang Truong and Kashu Yamazaki have these authors are contributed equally to this work.

✉ Khoa Vo
khoavoho@uark.edu

Sang Truong
sangt@uark.edu

Kashu Yamazaki
kyamazak@uark.edu

Bhiksha Raj
bhiksha@cs.cmu.edu

Minh-Triet Tran
tmtriet@hcmus.edu.vn

Ngan Le
thile@uark.edu

[1] AICV Lab, University of Arkansas, Fayetteville, Arkansas, USA

[2] Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

[3] University of Science, Ho Chi Minh City, Vietnam

[4] Vietnam National University, Ho Chi Minh City, Vietnam

action recognition (Kay et al., 2017), etc. In general, TAPG approaches can be divided into two main categories i.e. anchor-based approaches and boundary-based approaches. Inspired by anchor-based object detection in 2D images, anchor-based TAPG methods (Richard and Gall, 2016; Chao et al., 2018; Heilbron et al., 2016; Shou et al., 2016; Gao et al., 2017) pre-define a set of anchor segments and try to fit them into groundtruth action segments in videos. Even though a regression network has been applied to refine the proposals, anchor-based TAPG methods cannot fit all groundtruth actions with diverse lengths by a finite number of anchors. Boundary-based TAPG methods (Lin et al., 2018; Su et al., 2020; Lin et al., 2019, 2020; Xu et al., 2020; Vo-Ho et al., 2021; Vo et al., 2021) address the previous limitations by first separately localizing starting and ending timestamps of exiting actions and then fusing them by a follow-up action evaluation module.

Despite good achievements on benchmarking datasets, boundary-based approaches (Lin et al., 2018; Su et al., 2020; Lin et al., 2019, 2020; Xu et al., 2020) still have some limitations, in which the most major one is the overlooked video representation. In such designs, a video is split into consecutive snippets (or clips, chunks) of δ frames; then, a 3D convolutional backbone network (Ji et al., 2013; Carreira and Zisserman, 2017; Simonyan and Zisserman, 2014; Feichtenhofer et al., 2019) is simply applied to the entire spatial domain of each snippet to extract its visual representation. However, not all spatial regions in a snippet are relevant nor contribute to the formation of an action. Specifically, as shown in Fig. 1a, an actor itself rather than spatial environment influences the action i.e. jogging can be created anywhere regardless of environment. To address those limitations, Vo-Ho et al. (2021), Vo et al. (2021) recently propose to separately represent each snippet by both local actors features and global surrounding environment features. Both features are combined by a self-attention module to flexibly balance between local and global visual representation. Although the improvements reported (Vo-Ho et al., 2021; Vo et al., 2021) are very promising, those paradigms are unable to discriminate main actors who actually commit actions from the inessential actors, as shown in Fig. 1b. Additionally, both AEN and ABN may not be helpful in many videos where actions are not dependent on the presence of humans, i.e., egocentric videos, as shown in Fig. 1c.

Intuitively, besides actors and the environment, we, as human beings, also perceive an action through the presence of relevant objects and their interactions with the surroundings. However, unlike actors, relevant objects are often tiny with few pixels (e.g., less than 20×20 pixels). This causes the problem of vanishing information if we obtain objects from visual feature maps extracted by some typical CNN-based backbone. A possible solution is to leverage "vision and language" methods (Mei et al., 2020; Anderson et al., 2018; Radford et al., 2021) to represent objects by linguistic features. By this way, information about the presence of every object is fully preserved. We leverage CLIP (Radford et al., 2021), which is a powerful model to associate both vision and language, to extract linguistic features from relevant objects existing in video snippets. As a result, relevant objects are represented by linguistic features whereas environment and main actors are represented by visual features. Figure 1d illustrates our intuition for video representation.

In this paper, we propose a novel *Actors-Objects-Environment Interaction Network (AOE-Net)* as a simulation of human perception in modeling the video by visual features from main actors and environment as well as linguistic features from relevant objects. Our AOE-Net consists of two main modules, i.e., (i) Perception-based multi-modal representation (PMR) to extract visual-linguistic (V-L) feature and model actors-objects-environment relations in each snippet, (ii) Boundary-matching module (BMM) to localize and generate action proposals. To select only main actors along with choosing relevant objects and extract mutual relationships among each of these entities, we propose a novel *adaptive attention mechanism (AAM)*.

Our contributions are summarized as follows:

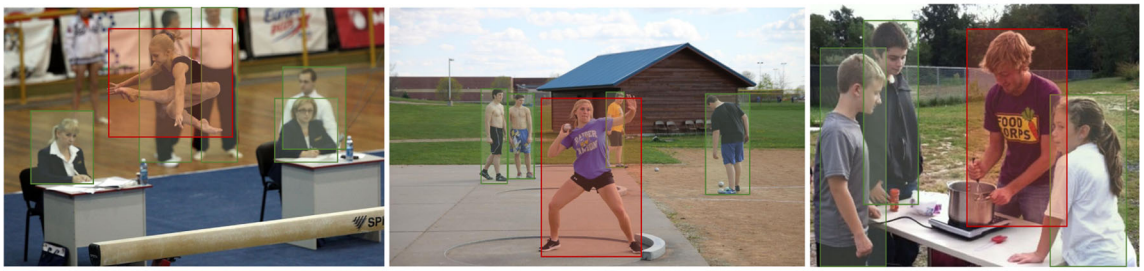
- We propose a novel network, AOE-Net, which follows the human perception process to understand human actions.
- We introduce a novel and effective attention module, AAM, which simultaneously selects main actors (or relevant objects) and eliminates inessential actor(s) (or objects), then, extracts semantic relations between main actors (or relevant objects).
- Our proposed AOE-Net achieves the SOTA performance on common benchmarking datasets of ActivityNet−1.3 and THUMOS-14 in both TAPG and TAD tracks with a large margin compared to previous works.
- We investigate the robustness of AOE-Net while working on egocentric videos of EPIC-KITCHENS 100, in which the main actor is absent from the video views.
- We provide ablation studies on the contribution of each entity type (i.e., main actors, relevant objects, and environment) as well as various combinations among them.
- Extensive ablation studies and qualitative analysis of AAM are also provided to investigate its effectiveness.

2 Related Works

2.1 Temporal Action Proposal Generation (TAPG)

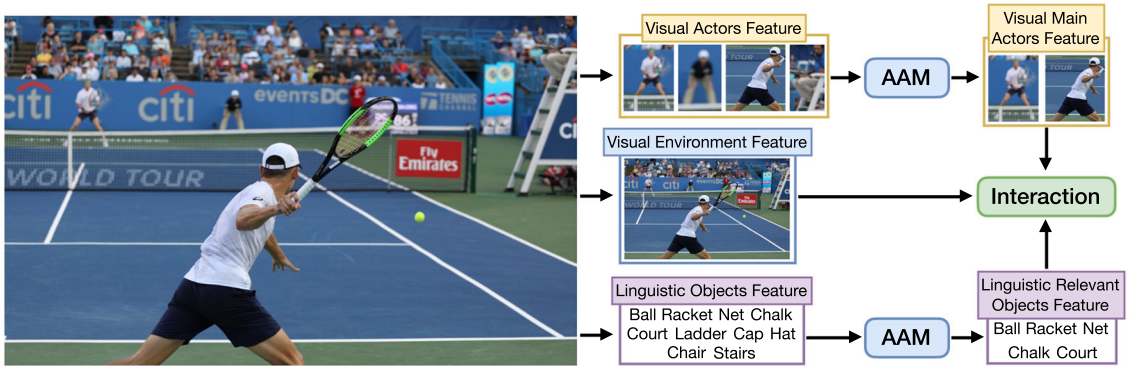
As stated above, prior works can be categorized into two groups: anchor-based and boundary-based. Anchor-based methods (Heilbron et al., 2016; Chao et al., 2018; Heil-

(a) Examples of actions (e.g jogging) are independent to environments.



(b) Examples of how actors contribute to form actions i.e. among all actors (green and red boxes) in the scenes, only main actors (red boxes) actually commit actions.

(c) Examples of actions in egocentric videos where actors are not visible.



(d) Our proposed AOE-Net is modeled by both global visual environment, local visual main actors features, linguistic relevant objects features, and the interaction among them. In AOE-Net, AAM is our proposed adaptive attention mechanism to select main actors and relevant objects.

Fig. 1 Most existing TAPG methods (Lin et al., 2018; Su et al., 2020; Lin et al., 2019, 2020; Xu et al., 2020) apply a 3D backbone network to entire spatial domain. However, as shown in **a**, actors contribute more importance to an action than environment itself. The SOTA in TAPG (Vo et al., 2021; Vo-Ho et al., 2021) extract both local humans features and global environment feature; however, they are unable to either distinguish between main actors who actually commit actions and inessential actors **b** or address egocentric videos where actors are not visible in the scene **c**. Our proposed AOE-Net, as illustrated in **d**, consists of the global visual environment, local visual main actors features, and linguistic relevant objects features

bron et al., 2016; Shou et al., 2016; Gao et al., 2017) are inspired by anchor-based object detection methods (Ren et al., 2015; Lin et al., 2017; Redmon and Farhadi, 2018), i.e., they pre-define a set of fixed segments and learn to fit them into groundtruth action segments in the video. Among them, Heilbron et al. (2016) uses space-time interest points and dictionary learning. Shou et al. (2016) makes use of C3D (Tran et al., 2015) to build a binary classification task to generate proposal segments. TURN (Gao et al., 2017) divides a video into units and employs unit-level features with a temporal regression. In some of those anchor-based approaches, a regression network is also applied to refine the proposals. However, the groundtruth proposals vary a lot in terms of duration, which discourages the performance of anchor-based methods. Boundary-based methods (Zhao et al., 2017; Lin et al., 2018; Su et al., 2020; Liu et al., 2019; Lin et al., 2019, 2020; Xu et al., 2020; Vo-Ho et al., 2021; Vo et al., 2021) resolve this problem by initially localizing the starting and ending timestamps of all actions appearing in the video and then matching them by an action evaluation module, which estimates the actionness score of every possible pair of boundaries. Among them, Zhao et al. (2017) adopts a watershed algorithm to group contiguous high-score as proposals. Lin et al. (2018) first predicts each temporal point as either starting or ending point of an action and then evaluates proposals. Gao et al. (2018a) combines both anchor-based and boundary-based approaches to reorganize the candidate set. Liu et al. (2019) employs a bilinear matching module to perform TAPG at two distinct granularities. Generally, the boundary-based methods provide superior performance over anchor-based methods.

Our AOE-Net belongs to the second category. Different from existing boundary-based methods, AOE-Net is based on V-L feature by leveraging the human perception principle.

2.2 Attention Module

Attention Models (Atts) have a long history (Itti et al., 1998) and have become an important concept in neural networks (Chaudhari et al., 2021). Atts can be divided into two main groups: Soft-Attention Models (Soft-Atts) and Hard-Attention Models (Hard-Atts). Bahdanau et al. (2014) was one of the first Soft-Atts that was applied to machine translation. Because of its differentiable architecture, which helps the whole model learn in an end-to-end fashion, Soft-Atts has become an essential component in a large number of applications (e.g., speech (Cho et al., 2015), NLP (Galassi et al., 2020), computer vision (Chaudhari et al., 2019)). Because self-attention networks (Vaswani et al., 2017) are able to learn the relations between input elements regardless of their quantity, their popularity is increasing in not only language models but also in computer vision. Hard-Atts was first introduced in Xu et al. (2015) and Elsayed et al. (2019) for digit and object classifications, respectively. Hard-Atts aims to mask out irrelevant elements of the inputs by sampling the input elements with probabilities to reduce the distractions. This is an advanced benefit over Soft-Atts; however, Hard-Att in Xu et al. (2015) is indifferentiable. Recently, Patro and Namboodiri (2018) proposes a Hard-Att that can be trained by normal gradient back-propagation, with a fundamental observation that the L2-norm values of more important features are usually higher than those of less important features in a feature map.

In this work, we propose an *adaptive attention model (AAM)*, which leverages both the differentiable Hard-Att (Patro and Namboodiri, 2018) and the self-attention network (Vaswani et al., 2017).

3 Our Method

Given an input video $\mathcal{V} = \{v_i\}_{i=1}^{N}$, where N is the number of frames, we follow the standard settings from existing works to divide $\mathcal{V}$ into a sequence of $\delta-$frame *snippets* s_i $|_{i=1}^{T}$. Each snippet s_i consists of δ consecutive frames, therefore, $\mathcal{V}$ has a total of $T = \lceil \frac{N}{\delta} \rceil$ snippets. Let $\phi(.)$ be an encoding function to extract the visual feature f_i of a δ-frame snippet s_i; the video $\mathcal{V}$ can be represented as $\mathcal{F}$ as follows:

$$\mathcal{F} = \{f_i\}_{i=1}^{T}, \text{ where } f_i = \phi(s_i) \tag{1}$$

Different from the existing works (Su et al., 2020; Lin et al., 2020; Long et al., 2019; Xu et al., 2020; Liu et al., 2020; Lin et al., 2019, 2018; Xu et al., 2020; Bai et al., 2020; Tan et al., 2021), which simply define $\phi(.)$ as a pre-trained backbone network (e.g., C3D (Ji et al., 2013), 2Stream (Simonyan and Zisserman, 2014), Slow-fast (Feichtenhofer et al., 2019)), we model $\phi(.)$ by the proposed PMR, which is capable of representing visual information of the snippet in both global and local perspectives, using both visual and linguistic information.

Given the feature sequence $\mathcal{F}$, the boundary-matching module (BMM) has a role of localizing action proposals. In this section, we introduce PMR in Sect. 3.1. Then, we present the boundary-matching module in Sect. 3.3.

3.1 Perception-based Multi-modal Representation (PMR)

PMR aims to extract features based on the principle of how a human perceives an action (i.e., identify the main actors at each temporal period, recognize relevant objects and understand interactions between main actors, relevant objects, and the environment) to specify when the action starts and ends. In this paper, we are interested in discovering two modalities of vision and language to extract V-L feature.

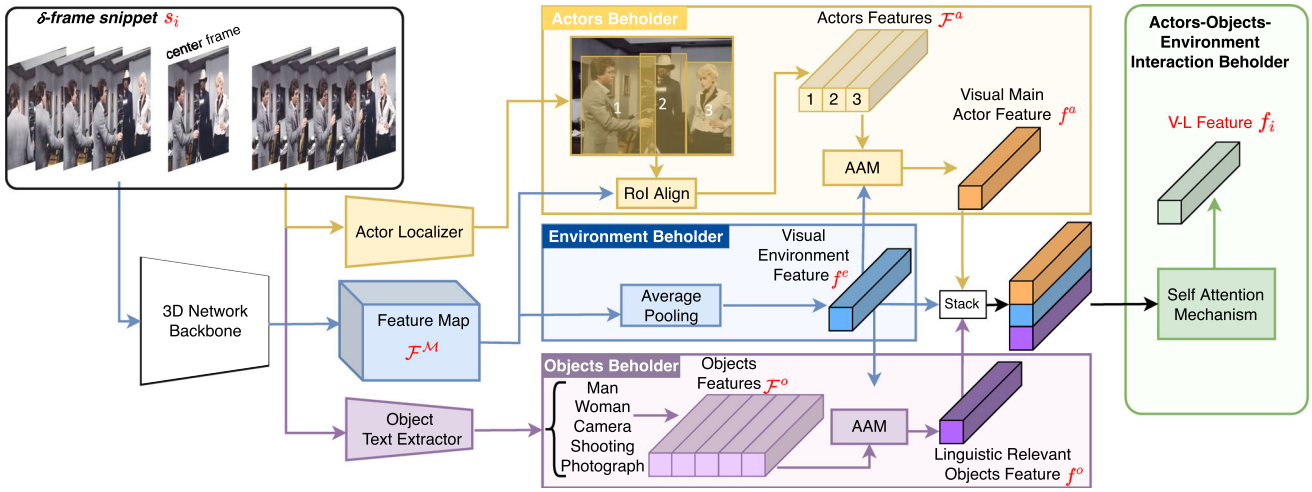


Fig. 2 The architecture of our proposed PMR. Given a δ-snippet s_i, the V-L feature is obtained by four modules: (i) Actors beholder to extract local visual action feature f^a; (ii) Environment beholder to extract global visual environment feature f^e; (iii) Objects beholder to extract linguistic object feature f^o, and (iv) Actors-objects-environment interaction beholder to model V-L feature as the interaction between actors, objects and the environment

PMR consists of four main components: (i) Environment beholder; (ii) Actors beholder; (iii) Objects beholder; and (iv) Actors-objects-environment interaction beholder. The overall architecture of PMR is shown in Fig. 2.

3.1.1 Environment Beholder

This component has the role of capturing the global visual information of an input δ-frame snippet. To extract the spatio-temporal information of the snippet, we adopt a 3D network pre-trained on action recognition benchmarking datasets as a backbone feature extractor. First, the snippet is processed through all convolutional blocks of the 3D network to obtain a feature map $\mathcal{F}^{\mathcal{M}}$ at the final block; then, an average pooling operator is employed to produce a spatio-temporal feature vector f^e.

3.1.2 Actors Beholder

This component semantically extracts visual main actors representation f^a. In most cases, an action cannot happen if a human (main actor) is absent notwithstanding environments (Fig. 1a). However, when an action occurs, it does not necessarily signal that every actor in the scene has committed the action (Fig. 1b). Herein, the actors beholder first localizes all existing actors in a δ-frame snippet. To do so, we apply a human detector onto the middle frame assuming that the actors would not move fast enough to be mislocated with a small δ. We denote $\mathcal{B} = \{b_i\}_{i=1}^{N_B}$ as a set of detected human bounding boxes, where $N_B \geq 0$. Afterwards, each of the detected bounding boxes, b_i, is aligned onto feature map $\mathcal{F}^{\mathcal{M}}$, which is obtained by the 3D network backbone from environment beholder, using RoIAlign (He et al., 2017). Then, each bounding box feature is average-pooled into a single feature vector f_i^a. Finally, we obtain a set of actor features $\mathcal{F}^a = \{f_i^a\}_{i=1}^{N_B}$.

To adaptively select an arbitrary number of main actors and extract their mutual relationships, we apply our proposed AAM (described in Sect. 3.2), which is elaborately explained in Sect. 3.2 and illustrated in Fig. 5.

3.1.3 Objects Beholder

Different from the environment and actors, objects may be tiny with a few pixels and therefore may vanish in the feature map $\mathcal{F}^{\mathcal{M}}$. Hence, in this objects beholder, we propose to use linguistic information from relevant objects, which is considerably more informative than visual information. We leverage CLIP (Radford et al., 2021) as a pre-trained model to extract linguistic information.

CLIP (Radford et al., 2021) is trained with a large number of image and description pairs, thus, CLIP effectively learns the correlation between the global scene information and local scene elements. Many scene elements are presented as small objects in the scene and they are hardly captured by an object detector. With CLIP, scene elements can be inferred by globally encoding the entire scene information. Thus, once the entire scene is captured, the small objects of scene elements are obtained accordingly.

For example, given an image of people playing tennis shown in Fig. 3 as below, it is unfeasible to detect a small object such as a tennis ball using an object detector. As shown in Fig. 3 (left), Mask-RCNN [49] is only able to

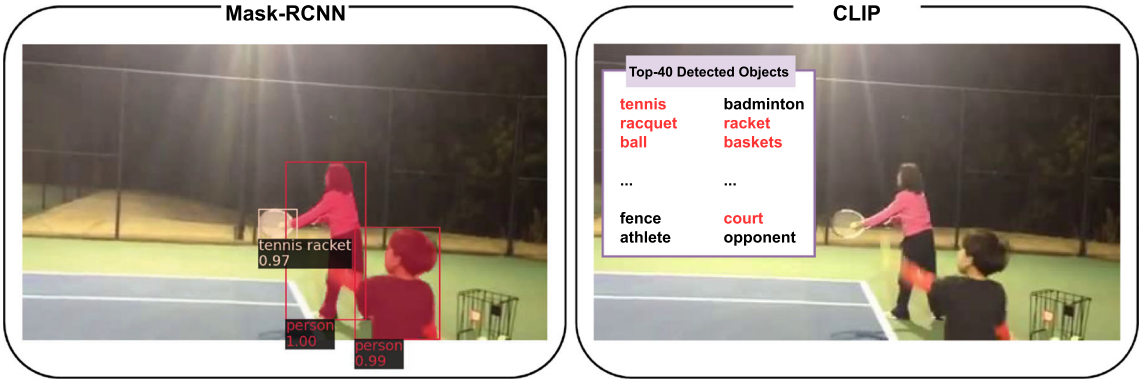


Fig. 3 An example of objects detected by Mask-RCNN (He et al., 2017) (left) vs. CLIP (right), where the most relevant objects selected by our AAM are highlighted in **bold red**) (Color figure online)

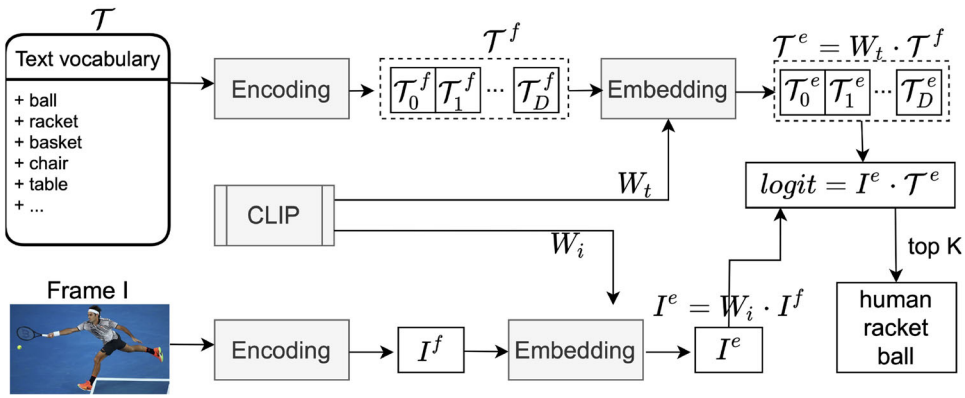


Fig. 4 Illustration of object text extraction where Encoding, Embedding, and CLIP are pre-trained models from Vaswani et al. (2017), Dosovitskiy et al. (2021), Radford et al. (2021), respectively

detect humans and tennis racket while the tennis ball is not captured. Whereas, CLIP already encoded tennis scene elements including tennis ball when modeling tennis games. As shown in Fig. 3 (right), CLIP captures tennis ball and other related objects such as basket, court, fence, etc. In this example, we choose top $K = 40$ detected objects by CLIP. The most relevant objects selected by AAM are shown in **bold red**.

Object text extraction is the first step of this component as illustrated in Fig. 4. However, our task just focuses on human activities and their related objects. Therefore, we utilize the corpus of ActivityNet Captioning dataset (Krishna et al., 2017) to construct the object text vocabulary $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{D}$.

ActivityNet Captioning dataset (Krishna et al., 2017) annotates the same set of videos in ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015). In its training split, there are a total of 37,447 sentences to densely describe every event in each video, these captions is composed by a vocabulary of up to 10,648 words. In order to create a vocabulary which majorly contains objects and human activities, we eliminate stop words, pronouns, numbers, and infrequent words (which appears 5 times or lower in the whole dataset). Afterwards, we remove words that do not present in the vocabulary used by CLIP (Radford et al., 2021). Fortunately, thanks to the byte pair (Sennrich et al., 2016) encoding used in CLIP (Radford et al., 2021), there are very few words that are removed after in this step. To this end, the vocabulary for our objects beholder consists of $D = 3,544$ words is extracted from the ActivityNet Captioning dataset (Krishna et al., 2017).

Each word $\mathcal{T}_i \in \mathcal{T}$ is encoded by a Transformer network (Vaswani et al., 2017) into a text feature $\mathcal{T}_i^f$. Let W_t be a text projection matrix pre-trained by CLIP, the embedding text vocabulary is computed as $\mathcal{T}^e = W_t \cdot \mathcal{T}^f$, where $\mathcal{T}^f = \{\mathcal{T}_i^f\}_{i=1}^{D}$. Let W_i be an image projection matrix pre-trained by CLIP, a middle frame I of the δ-frame snippet is first encoded by Vision Transformer (Dosovitskiy et al., 2021) to extract visual feature I^f, and then embedded by W_i, i.e., $I^e = W_i \cdot I^f$. The pairwise cosine similarities between embedded I^e and $\mathcal{T}^e$ is then computed. Top K similarity scores are chosen as output objects text represented by feature $\mathcal{F}^o = \{\mathcal{T}_i^f\}_{i=1}^{K}$. Ablation study on K will be discussed in Sect. 4.5.3. Similar to the actors beholder, we apply the proposed AAM (described in Sect. 3.2) to select relevant objects from $\mathcal{F}^o$, then model the semantic relations among them, and finally obtain linguistic feature f^o.

Algorithm 1 AAM to extract the representation of main actors in a snippet.

Data: Feature vector f^e and features set $\mathcal{F}^a$ represent environment and all actors that appear in an input snippet, respectively.
Result: Feature vector f^a represents main actors.

$\hat{f}^e \leftarrow MLP_{\theta_e}(f^e)$
set $\tilde{\mathcal{F}}^a$, H^a to empty list ▷ $\mathcal{F}^a$ stores selected main actors, H^a stores scores of every actor
for each f_i^a in $\mathcal{F}^a$ **do**
  $\hat{f}_i^a \leftarrow MLP_{\theta_a}(f_i^a)$
  $h_i^a \leftarrow || \hat{f}_i^a \oplus \hat{f}^e ||_2$ ▷ $\oplus$: element-wise addition
  append h_i^a to H^a
end for
$H^a \leftarrow softmax(H^a)$
$\tau \leftarrow \frac{1}{|\mathbf{h}^a|}$
for each h_i^a in H^a **do**
  if $h_i^a > \tau$ **then**
    append f_i^a to $\tilde{\mathcal{F}}^a$
  end if
end for
$f^a \leftarrow self\_attention(\tilde{\mathcal{F}}^a)$

3.1.4 Actors-Objects-Environment (AOE) Beholder

This component aims to model the relations between global visual environment feature f^e, local visual of main actors features f^a, and linguistic relevant objects features f^o. Firstly, we stack three types of features together as $\mathcal{F}^{aoe} = [f^a, f^o, f^e]$. Then, we employ the self-attention model (Vaswani et al., 2017) followed by an average pooling layer to fuse the stack of features $\mathcal{F}^{aoe}$ into f_i. f_i is a V-L feature that represents the input snippet s_i through both visual (environment and actors modalities) and linguistic (objects modality) ways.

3.2 Adaptive Attention Mechanism (AAM)

Given M actors (or objects) obtained in the input snippet, only a few of those, i.e., $\hat{M}$ main actors (or relevant objects), actually contribute to an action. Because $\hat{M}$ is unknown and continuously changes throughout the input video, we propose AAM that inherits the merits from adaptive hard attention (Malinowski et al., 2018) to select an arbitrary number of main actors (or objects) and a soft self-attention mechanism (Vaswani et al., 2017) to extract relationships among them. Take actors beholder as an instance, AAM is described by the pseudocode in Algorithm 1 and illustrated in Fig. 5.

To begin, the environment feature f^e and actors features $\mathcal{F}^a$ are embedded into the same dimensional space by a multi-layer perceptrons (MLPs) parameterized by θ_e and θ_a, respectively:

$$\hat{f}^e = MLP_{\theta_e}(f^e) \tag{2}$$

$$\hat{F}^a = \{\hat{f}_i^a\}_{i=1}^M \text{ where } \hat{f}_i^a = MLP_{\theta_a}(f_i^a) \tag{3}$$

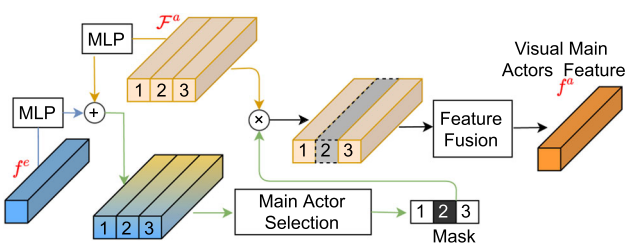


Fig. 5 Illustration of proposed AAM. We choose actors features F^a and environment feature f^e as an example. AAM aims to select main actors features, followed by fusing arbitrary main actors features, to obtain visual main actors representation f^a

Then, $\hat{f}^e$ is combined with each feature $\hat{f}_i^a$ of $\hat{F}^a$ by element-wise addition (i.e., $\oplus$) to form a collaborative feature. Afterwards, we can compute the attention score h_i^a corresponding to $\hat{f}_i^a$ using the L2-norm of its corresponding collaborative feature. These computational steps can be presented through the following equation:

$$h_i^a = || \hat{f}_i^a \oplus \hat{f}^e ||_2 \tag{4}$$

It is proven in Malinowski et al. (2018) that features with the greater L2-norm values carry more meaningful information and better contribute to later modules.

Next, we re-scale all L2-norm values by softmax function to be summed up to 1.0, because L2-norm values are unbounded:

$$H^a = \{h_i^a\}_{i=1}^M, \text{ where } h_i^a = \frac{e^{h_i^a}}{\Sigma_{i=1}^M e^{h_i^a}} \tag{5}$$

To obtain the features of an arbitrary number of main actors, we create an adaptive threshold based on the total number of actors $\tau = \frac{1}{|\mathcal{F}^a|}$ and retrieve only features $f_i^a \in \mathcal{F}^a$ with corresponding score higher than τ:

$$\tilde{\mathcal{F}}^a = \{f_i^a \mid h_i^a \geq \tau\} \tag{6}$$

After that, we fuse a set of main actors feature vectors $\tilde{\mathcal{F}}^a$ into a single feature vector f^a by leveraging the self-attention Transformer Encoder proposed in Vaswani et al. (2017).

In the case of objects beholder, the input actors features $\mathcal{F}^a$ is replaced by the objects features $\mathcal{F}^o$.

3.3 Boundary-Matching Module (BMM)

BMM is responsible for localizing action boundary and generating action proposals in videos. In our AOE-Net, BMM module is adopted from previous works i.e. BSN (Lin et al., 2018), BMN (Lin et al., 2019), ABN (Vo-Ho et al., 2021), AEN (Vo et al., 2021), AEI (Vo et al., 2021) because of its standard and simple design. BMM takes the output V-L features sequence $\mathcal{F} = \{f_i\}_{i=1}^T$ from PMR module as its input.

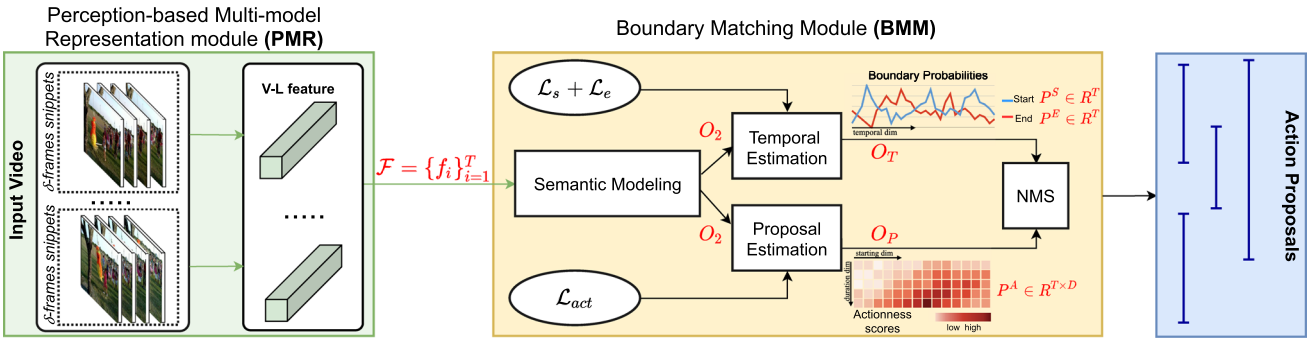


Fig. 6 The overall architecture of our proposed AOE-Net, consisting of perception-based multi-model representation module (PMR) and boundary-matching module (BMM)

Table 1 The detailed architecture of BMM with three components.

Layers	Input	Output
1DConv. 256 × 3/1, ReLU	$\mathcal{F} : F \times T$	$O_1 : 256 \times T$
1DConv. 128 × 3/1, ReLU	$O_1 : 256 \times T$	$O_2 : 128 \times T$
1DConv. 256 × 3/1, ReLU	$O_2 : 128 \times T$	$O_3 : 256 \times T$
1DConv. 2 × 3/1, Sigmoid	$O_3 : 256 \times T$	$O_T : 2 \times T$
Matching layer	$O_2 : 128 \times T$	$O_5 : 128 \times 32 \times D \times T$
3DConv. 512 × 32 × 1 × 1/(32, 0, 0), ReLU	$O_5 : 128 \times 32 \times D \times T$	$O_6 : 512 \times 1 \times D \times T$
Squeeze	$O_6 : 512 \times 1 \times D \times T$	$O_7 : 512 \times D \times T$
2DConv. 128 × 1 × 1/(0, 0), ReLU	$O_7 : 512 \times D \times T$	$O_8 : 128 \times D \times T$
2DConv. 128 × 3 × 3/(1, 1), ReLU	$O_8 : 128 \times D \times T$	$O_9 : 128 \times D \times T$
2DConv. 2 × 1 × 1/(0, 0), Sigmoid	$O_9 : 128 \times D \times T$	$O_P : 1 \times D \times T$

$\mathcal{F}$ is the input feature obtained from PMR. T and D are the temporal length of the video and maximum duration of proposals in terms of the number of snippets

Our BMM contains three components: semantic modeling, temporal estimation (TE), and proposal estimation (PE) as illustrated in Fig. 6. The first component models the semantic relationship between snippets. The TE component assesses each snippet $s_i \ |_{i=1}^{T}$ to evaluate probabilities of action starting (P_i^S) and action ending (P_i^E) that exist in s_i. Meanwhile, the PE component evaluates every interval $[i, j]$ in the video to estimate its actionness score $P_{i,d}^A$, where $d = j - i$. The detailed architecture of BMM is provided in Table 1. The semantic modeling component is implemented by two 1-D Conv. layers and outputs a feature map $O_2 \in R^{128 \times T}$. The later components, TE and PE, take O_2 as their input and generate $O_T \in R^{2 \times T}$ and $O_P \in R^{1 \times D \times T}$, respectively. The output O_T presents probabilities of action starts ($P^S \in R^T$) and action ends ($P^E \in R^T$). The output O_P contains actionness scores $P^A \in R^{D \times T}$.

At the inference stage, we search through P^S and P^E to select temporal locations i whose P_i^S or P_i^E are local maximums to form sets of potential starting and ending temporal locations, respectively. Then, starting and ending locations (s, e) (e.g. $s \leq e \leq T$) are paired and become a candidate proposal with the score $s = P_s^S \cdot P_e^E \cdot P_{s,e-s}^A$. Based on the timestamps and scores of candidate proposals, we finally apply NMS (Bodla et al., 2017; Neubeck and Van Gool, 2006) to produce the final set of temporal action proposals.

3.4 Training Methodology

3.4.1 Training Labels Generation from Groundtruth

We follow (Lin et al., 2019, 2018) to generate the ground truth labels for training process including starting labels, ending labels for $\mathcal{L}_s$, $\mathcal{L}_e$ and duration labels for $\mathcal{L}_{act}$.

The starting and ending labels are generated for every snippet of the video, which are called $L_S = \{l_n^s\}_{n=1}^T$ and $L_E = \{l_n^e\}_{n=1}^T$, respectively. The boundary timestamps (starting and ending) of every action instance $a_i = (s_i, e_i)$ are rescaled into T-snippet range by multiplying them with $\frac{T \cdot \text{fps}}{L}$ where fps is the frame rate of the video and the action instance $a_i \in \mathcal{A}$, $\mathcal{A} = \{a_i\}_{i=1}^M$. After rescaling, the action instance a_i becomes a new action instance $a_i^\delta = (s_i^\delta, e_i^\delta)$. For every snippet $t_n \in T$, we denote a temporal region $r_n = [t_n - 1, t_n + 1]$. Analogously, for every pair of boundaries (s_i^δ, e_i^δ) of action a_i^δ, we denote regions $r_i^s = [s_i^\delta - \frac{3}{2}, s_i^\delta + \frac{3}{2}]$ and $r_i^e =$

$[e_i^\delta - \frac{3}{2}, e_i^\delta + \frac{3}{2}]$ as their corresponding starting region and ending region. By this formulation, we have two sets of regions $R_S = \{r_i^s\}_{i=1}^M$ and $R_E = \{r_i^e\}_{i=1}^M$ for starting and ending boundaries, respectively. Finally, starting label l_n^s and ending label l_n^e of a snippet t_n are calculated by the following functions:

$$l_n^{\mathrm{s}} = \begin{cases} 1, & \sum_{i=1}^{M} \frac{|r_n \cap r_i^{\mathrm{s}}|}{|r_i^{\mathrm{s}}|} \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \qquad l_n^{\mathrm{e}} = \begin{cases} 1, & \sum_{i=1}^{M} \frac{|r_n \cap r_i^{\mathrm{e}}|}{|r_i^{\mathrm{e}}|} \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

The duration labels for a video are gathered into a matrix $L_D \in \{0, 1\}^{D \times T}$ where D is the maximum length of proposals being considered in number of snippets, as suggested in Lin et al. (2019), we set $D = T$ in all of our experiments. With an element at position (t_i, t_j) stands for a proposal action $a_p = (t_s = \frac{t_j \cdot T}{t_v}, t_e = \frac{(t_j + t_i) \cdot T}{t_v})$, it will be assigned by 1 if its Interaction-over-Union with any ground truth action in $\mathcal{A} = \{a_i\}_{i=1}^M$ reaches a local maximum, or 0 otherwise.

3.4.2 Loss Functions

To train our AOE-Net with the groundtruth labels, we define the loss function $\mathcal{L}_{AOE}$ as in Eq. (7) where $\mathcal{L}_s$, $\mathcal{L}_e$, and $\mathcal{L}_{act}$ are loss functions corresponding to starting boundary, ending boundary and actionness score.

$$\mathcal{L}_{AOE} = \mathcal{L}_s(P^S, L^S) + \mathcal{L}_e(P^E, L^E) + \mathcal{L}_{act}(P^A, L^A) \quad (7)$$

We use weighted binary log-likelihood loss $\mathcal{L}_{wb}$ for $\mathcal{L}_s$ and $\mathcal{L}_e$, which is defined as follows:

$$\mathcal{L}_{wb}(P, L) = \sum_{i=1}^{N} \left[\frac{L_i}{N^+} \log P_i + \frac{(1 - L_i)}{N^-} \log(1 - P_i) \right] \quad (8)$$

where N^+ and N^- are the number of positives and negatives in groundtruth labels, respectively. Conversely, $\mathcal{L}_{act}(P, L)$ is defined as follows:

$$\mathcal{L}_{act}(P, L) = \mathcal{L}_{wb}(P, L) + \lambda \mathcal{L}_2(P, L), \quad (9)$$

where $\mathcal{L}_2$ is the mean squared error loss and λ is set to 10.

To reduce time cost in the training phase of our proposed AOE-Net, actors features $\mathcal{F}^a$, objects features set $\mathcal{F}^o$ and environment feature f^e are extracted in advance. Then, AAM and AOE Interaction beholder of PMR module is trained with BMM module in an end-to-end framework.

4 Experiments

4.1 Datasets and Metrics

4.1.1 Datasets

Our experiments on TAPG and TAD are carried out using both ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015) and THUMOS-14 (Jiang et al., 2014) datasets. The former features 20K videos and 200 activities that have been annotated, whereas the latter has 414 videos and 20 types of actions. We follow prior works (Lin et al., 2018, 2019, 2020) for videos preprocessing with the snippet length set to $\delta = 16$ in all experiments. To prove the effectiveness of our proposed AOE-Net on egocentric videos, we also conduct an experiment on TAPG task of EPIC-KITCHENS 100 dataset (Damen et al., 2021), which consists of 100 video hours, 20 M frames, 90K actions in 700 variable-length videos captured in 45 environments using head-mounted cameras.

4.1.2 Metrics

In TAPG, we use two common metrics, i.e., AR@AN and AUC, to evaluate the proposed AOE-Net as well as compare it with SOTA approaches. The former metric is the average recall (AR) calculated at a specific average number of proposals (AN) preserved by each video. The latter one is the area under the AR versus the AN curve score. AR@100 and AUC are the most commonly used metrics in ActivityNet−1.3. In THUMOS-14, however, just AR@AN is utilized to compare approaches; nonetheless, multiple AN are chosen from a list of [50, 100, 200, 500, 1000].

In TAD, we use mean Average Precision (mAP) to benchmark approaches. Following the common settings (Lin et al., 2018, 2020, 2019; Liu et al., 2020; Zhao et al., 2020), we evaluate TAD methods in ActivityNet−1.3 with tIoU thresholds of {0.5, 0.75, 0.95}, and average mAP. Whereas, TAD methods in THUMOS-14 are evaluated with tIoU thresholds of {0.3, 0.4, 0.5, 0.6, 0.7}.

4.2 Implementation Details

To extract visual features from videos, we use a C3D (Ji et al., 2013) network pre-trained on Kinetics-400 (Kay et al., 2017) as the backbone network in all experiments on both ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015) and THUMOS-14 (Jiang et al., 2014) (unless stated otherwise). The dimensions of the features extracted from the C3D backbone are 2048.

In the objects beholder, to extract object text, we use CLIP (Radford et al., 2021) that was pre-trained on a large-scale dataset of 400 M image-text pairs crawled from the Internet. The text feature and image feature are encoded by

Table 2 **TAPG** comparisons on ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015) in terms of AR@100 and AUC on validation set and AUC on testing set

Methods	Venue & Year	Feature	AR@100	AUC(val)	AUC(test)
TCN Dai et al. (2017)	ICCV17	2Stream	–	59.58	61.56
MSRA Yao et al. (2017)	CVPRW17	P3D	–	63.12	64.18
SSTAD Buch et al. (2017)	BMVC17	C3D	73.01	64.40	64.80
CTAP Gao et al. (2018a)	ECCV18	2Stream	73.17	65.72	–
BSN Lin et al. (2018)	ECCV18	2Stream	74.16	66.17	66.26
SRG Eun et al. (2019)	IEEE-TCSVT19	2Stream	74.65	66.06	–
MGG Liu et al. (2019)	CVPR19	I3D	74.54	66.43	66.47
BMN Lin et al. (2019)	ICCV19	2Stream	75.01	67.10	67.19
DBG Lin et al. (2020)	AAAI20	2Stream	76.65	68.23	68.57
BSN++ Su et al. (2020)	ACCV20	2Stream	76.52	68.26	–
TSI++ Liu et al. (2020)	ACCV20	2Stream	76.31	68.35	68.85
MRZhao et al. (2020)	ECCV20	I3D	75.27	66.51	–
AEN Vo-Ho et al. (2021)	ICASSP21	C3D	75.65	68.15	68.99
ABN Vo et al. (2021)	IEEE-Access21	C3D	76.72	69.16	69.26
SSTAP Wang et al. (2021)	CVPR21	I3D	75.54	67.53	–
TCANet Qing et al. (2021)	CVPR21	2Stream	76.08	68.08	–
Zheng, et.al. Zheng et al. (2021)	NPL21	2Stream	74.93	65.20	–
AEI Vo et al. (2021)	BMVC21	C3D	*77.24*	*69.47*	*70.09*
AOE-Net		C3D	**77.67**	**69.71**	**70.10**
		2Stream	76.32	68.35	69.00
		SlowFast	76.95	68.95	69.86

Transformer (Vaswani et al., 2017) and Vision Transformer (Dosovitskiy et al., 2021) networks, respectively. In the actors beholder, to detect humans, we use a Faster-RCNN model (Ren et al., 2015) that has been pre-trained on the COCO dataset (Lin et al., 2014). Adam optimizer was used to train our AOE-Net, and the initial learning rate is set to 0.0001 for ActivityNet−1.3 and 0.001 for THUMOS-14.

On ActivityNet−1.3, Soft-NMS (SNMS) (Bodla et al., 2017) is used in post-processing for all experiments in TAPG and TAD. On THUMOS-14, following (Lin et al., 2018, 2019), both Soft-NMS (Bodla et al., 2017) and NMS (Neubeck and Van Gool, 2006) are utilized in post-processing of TAPG, whereas only NMS is applied in TAD. In the following experimental results, we emphasize the best performance in **bold** and the second-best performance in underline.

4.3 Performance and Comparison on TAPG

Table 2 presents TAPG comparison on both validation and testing sets of ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015). The experimental results demonstrate that our approach AOE-Net with C3D (Ji et al., 2013) feature outperforms the existing methods in terms of AR@100 and AUC by an adequate margin. Table 3 shows the TAPG comparison on THUMOS-14. Compared to the existing TAPG methods, our AOE-Net performs very competitive on AR@ANs metrics with both SNMS and NMS. On SNMS, AOE-Net obtains the second best on all AR@ANs, except AR@100 where it is competitive to the best ones (50.26 vs. 50.67). On NMS, AOE-Net obtains the best on AR@100 and the second best on AR@200 and AR@500 with very close gap with the SOTA, 57.49 vs. 57.74 and 62.40 vs. 62.74, respectively. Notably, the performance on TAPG in both datasets of our AOE-Net are a very competitive with AEI-B (Vo et al., 2021) and followed closely by ABN (Vo et al., 2021), both of which also incorporate local actors and global environment. This experiment strongly supports our observation and motivation on using the human perception principle to analyze human actions in untrimmed videos.

Beside solely evaluating AOE-Net on TAPG and TAD tasks, the effects of different backbone features to our AOE-Net also worth an investigation. The performance of our proposed AOE-Net network on different features, i.e., C3D (Ji et al., 2013), 2Stream (Simonyan and Zisserman, 2014) and Slowfast (Feichtenhofer et al., 2019), with the features dimensions are 2048, 2314 and 400, respectively, are reported in the bottom part of Table 2 on TAPG task of ActivityNet−1.3 dataset (Fabian Caba Heilbron and Niebles, 2015). As demonstrated, we notice that the performance with C3D (Ji et al., 2013) features are state-of-the-art, while the

Table 3 **TAPG** comparisons on **THUMOS-14** in terms of AR@AN, where SNMS represents Soft-NMS (Bodla et al., 2017)

	Methods	Venue & Year	Feature	@50	@100	@200	@500	@1000	Average
SNMS	CTAP Gao et al. (2018a)	ECCV18	2Stream	32.49	42.61	51.97	–	–	–
	BSN Lin et al. (2018)	ECCV18	2Stream	37.46	46.06	53.21	60.64	64.52	52.38
	MGG Liu et al. (2019)	CVPR19	I3D	39.93	47.75	54.65	61.36	64.06	53.55
	BMN Lin et al. (2019)	ICCV19	2Stream	39.36	47.72	54.70	62.07	65.49	53.87
	DBG Lin et al. (2020)	AAAI20	2Stream	37.32	46.67	54.50	62.21	66.40	53.42
	Rapnet Gao et al. (2020)	AAAI20	C3D	40.35	48.23	54.92	61.41	–	–
	TSI++Liu et al. (2020)	ACCV20	2Stream	42.30	*50.51*	57.24	63.43	–	–
	MRZhao et al. (2020)	ECCV20	I3D	44.23	**50.67**	55.74	–	–	–
	BC-GNN Bai et al. (2020)	ECCV20	2Stream	40.50	49.60	56.33	62.80	–	–
	TCANet Qing et al. (2021)	CVPR21	2Stream	42.05	50.48	57.13	63.61	66.88	*56.03*
	SSTAP Wang et al. (2021)	CVPR21	2Stream	41.01	50.12	56.69	–	**68.81**	–
	ABN Vo et al. (2021)	IEEE-Access21	C3D	40.87	49.09	56.24	63.53	67.29	55.40
	AEI-B Vo et al. (2021)	BMVC21	C3D	**44.97**	50.13	**57.34**	**64.43**	67.78	**56.93**
	AOE		C3D	*44.56*	50.26	*57.30*	*64.32*	*68.19*	**56.93**
NMS	BSNLin et al. (2018)	ECCV18	C3D	27.19	35.38	43.61	53.77	59.50	43.89
	BSNLin et al. (2018)	ECCV18	2Stream	35.41	43.55	52.23	61.35	*65.10*	51.53
	BMNLin et al. (2019)	ICCV19	C3D	29.04	37.72	46.79	56.07	60.96	46.12
	BMNLin et al. (2019)	ICCV19	2Stream	37.15	46.75	54.84	62.19	**65.22**	53.23
	DBGLin et al. (2020)	AAAI20	C3D	32.55	41.07	48.83	57.58	59.55	47.92
	DBGLin et al. (2020)	AAAI20	2Stream	40.89	49.24	55.76	61.43	61.95	53.85
	ABNVo et al. (2021)	IEEE-Access21	C3D	*44.89*	51.86	57.36	61.67	62.59	55.67
	AEI-B Vo et al. (2021)	BMVC21	C3D	**45.74**	*52.39*	**57.74**	**62.49**	63.38	**56.35**
	AOE		C3D	44.78	**52.41**	*57.49*	*62.40*	63.40	*56.10*

performance with SlowFast (Feichtenhofer et al., 2019) features are closely behind. Whereas, the performance with 2Stream (Simonyan and Zisserman, 2014) features is the worst in three types of backbone features.

In TAPG, *generalizability* is also a significant criterion to evaluate a method. Following the same experiment setup in Lin et al. (2018), Lin et al. (2019), Lin et al. (2020), Liu et al. (2020), Vo et al. (2021), we conduct this study on ActivityNet−1.3 with two subsets, i.e., *Seen*: "Sports, Exercises, and Recreation" and *Unseen*: "Socializing, Relaxing, and Leisure". Our AOE-Net is trained on *Unseen*+*Seen* and *Seen* training sets, separately, and then evaluated on the *Seen* and *Unseen* validation sets. Figure 7 provides the performance comparison and visualization between AOE-Net with other SOTA methods. In each chart on the right, the performance of AOE-Net is shown in the last columns, which demonstrates that AOE-Net is superior to other SOTA methods. Figure 7 also shows that our AOE-Net achieves good performances on *Seen* validation set with an acceptable drop on *Unseen* validation set on both training configurations, suggesting that our AOE-Net is highly generalizable to unseen action types.

4.4 Performance and Comparison on TAD

For a fair comparison, we follow the experiment settings in Lin et al. (2018), Lin et al. (2019), Lin et al. (2020), Xu et al. (2020), Bai et al. (2020), Liu et al. (2019), Tan et al. (2021), Vo et al. (2021) to produce labels for action proposals produced by our AOE-Net. On AcitivityNet−1.3, we adopt the top-1 video-level classification results generated by the method in Xiong et al. (2016) for our proposals. Whereas on THUMOS-14, we instead label our action proposals with either UntrimmedNet (Wang et al., 2017) (top-2 classification results) or P-GCN (Zeng et al., 2019).

Table 4 shows TAD performance comparison between AOE-Net and other SOTA methods on ActivityNet−1.3 validation set. The results emphasize that our method outperforms SOTA methods on multiple tIoU thresholds. The experiment results on THUMOS-14 test set in Table 5 demonstrate that our AOE-Net is superior to other SOTA methods on most of the metrics with both classifiers.

4.5 Ablation Study

We further conduct a rich ablation study to show the effectiveness of each component in the proposed AOE-Net as well

Methods	Training	Evaluation			
		Seen		Unseen	
		AR@100	AUC	AR@100	AUC
BSN [1]	Seen + Unseen	72.40	63.80	71.84	63.99
	Seen	72.42	64.02	71.32	63.38
BMN [3]	Seen + Unseen	72.96	65.02	72.68	65.05
	Seen	72.47	64.37	72.46	64.47
TSI++ [45]	Seen + Unseen	74.69	66.54	74.31	66.14
	Seen	73.59	65.60	73.07	65.05
DBG [4]	Seen + Unseen	73.30	66.57	67.23	64.59
	Seen	72.95	66.23	64.77	62.18
ABN [6]	Seen + Unseen	74.58	66.96	75.25	67.49
	Seen	74.40	66.69	73.66	65.49
AOE-Net	Seen + Unseen	76.36	68.31	77.31	69.07
	Seen	76.43	68.42	74.90	66.92

Fig. 7 *Generalizability* evaluation and comparisons on Activity Net−1.3 in terms of AR@100 and AUC. Methods are trained on *Unseen+Seen* and *Seen* training sets, respectively; and are evaluated on *Seen* (first two charts) and *Unseen* (last two charts) validation sets. Top: Detailed performance of individual experiment setting of various methods. Bottom: Visualized generalizability comparison between our proposed AOE-Net and other methods

Table 4 **TAD** comparisons on **ActivityNet−1.3** in terms of mAP@tIoU and mAP, where the proposals are combined with video-level classification results generated by Xiong et al. (2016)

Methods	Venue & Year	Feature	0.50	0.75	0.95	Average
BSN Lin et al. (2018)	ECCV18	2Stream	46.45	29.96	8.02	30.03
GTAN Long et al. (2019)	CVPR19	P3D	*52.61*	34.14	8.91	34.31
BMN Lin et al. (2019)	ICCV19	2Stream	50.07	34.60	8.29	33.85
GTAD Xu et al. (2020)	CVPR20	2Stream	50.36	34.60	9.02	34.09
P-GCN Zeng et al. (2019)	CVPR20	I3D	42.90	28.14	2.47	26.99
MR Zhao et al. (2020)	ECCV20	2Stream	43.47	33.91	9.21	30.12
TSI++ Liu et al. (2020)	ACCV20	2Stream	51.18	*35.00*	6.59	34.15
BC-GNN Bai et al. (2020)	ECCV20	2Stream	50.56	34.75	9.37	34.26
RTD Tan et al. (2021)	ICCV21	2Stream	47.21	30.68	8.61	30.83
ABN Vo et al. (2021)	IEEE-Access21	C3D	51.78	34.18	*10.29*	34.22
AEI-B Vo et al. (2021)	BMVC21	C3D	52.3	34.5	9.7	**34.7**
AOE		C3D	**54.42**	**35.43**	**10.35**	*34.48*

as the robustness of AOE-Net to egocentric videos. We also report the network efficiency and AOE-Net performance with different settings of hyper-parameter K. Additional ablation study will be included in supplementary.

4.5.1 Contribution of each Beholder

We examine TAPG performance on THUMOS-14 with different network settings as given in Table 6. While the performance of each individual beholder is shown in the Exps.#1-3, different combinations of features are given in Exps.#4-7. This emphasizes the important contribution of

Table 5 **TAD** comparisons on **THUMOS-14** in term of mAP@tIoU using two different classifiers, i.e., UntrimmedNet (Wang et al., 2017) and P-GCN (Zeng et al., 2019)

	Methods	Year	Feature	0.7	0.6	0.5	0.4	0.3	Average
UntrimmedNet Wang et al. (2017)	BSN Lin et al. (2018)	ECCV18	2Stream	20.0	28.4	36.9	45.0	53.5	36.76
	BMN Lin et al. (2019)	ICCV19	2Stream	20.5	29.7	38.8	47.4	56.0	38.48
	MGG Liu et al. (2019)	CVPR19	2Stream	21.3	29.5	37.4	46.8	53.9	37.78
	GTAN Long et al. (2019)	CVPR19	P3D	–	–	38.8	47.2	57.8	–
	DBG Lin et al. (2020)	AAAI20	2Stream	21.7	30.2	39.8	49.4	57.8	39.78
	GTAD Xu et al. (2020)	CVPR20	2Stream	23.4	30.8	40.2	47.6	54.5	39.30
	TSI++Liu et al. (2020)	ACCV20	2Stream	22.4	33.2	42.6	52.1	*61.0*	42.26
	BC-GNN Bai et al. (2020)	ECCV20	2Stream	23.1	31.2	40.4	49.1	57.1	40.18
	BU-TALZhao et al. (2020)	ECCV20	2Stream	**28.5**	*38.0*	45.4	50.7	53.9	43.30
	TCANet Qing et al. (2021)	CVPR21	2Stream	26.7	36.8	44.6	53.2	60.6	44.38
	RTDTan et al. (2021)	ICCV21	2Stream	25.0	36.4	45.1	53.1	58.5	43.62
	ABN Vo et al. (2021)	IEEE-Access21	C3D	25.6	37.0	*46.1*	*54.0*	59.9	44.51
	AEI-B Vo et al. (2021)	BMVC21	C3D	23.4	35.9	44.7	52.7	58.7	43.08
	AOE-Net	–	C3D	*25.8*	**38.8**	**48.4**	**57.3**	**63.4**	**46.74**
P-GCN Zeng et al. (2019)	BSNLin et al. (2018)	ECCV18	I3D	–	–	49.1	57.8	63.6	–
	MRZhao et al. (2020)	ECCV20	2Stream	–	–	50.10	60.99	66.29	–
	GTAD Xu et al. (2020)	CVPR20	2Stream	22.9	37.6	**51.6**	*60.4*	*66.4*	*47.78*
	AOE-Net	–	C3D	**23.5**	37.4	*50.9*	**60.6**	**67.1**	**47.89**

Table 6 TAPG comparisons on different network settings. Act., Env., Obj. denote actors, environment, objects beholders

Exp	Setting					TAPG Performance				
	Act.	Env.	Obj.	AAM	Soft-Att	@50	@100	@200	@500	@1000
#1	✓	✗	✗	✗	✓	25.96	35.14	43.48	52.37	58.47
#2	✗	✓	✗	✗	✗	38.94	47.80	54.93	61.92	65.96
#3	✗	✗	✓	✗	✓	18.06	26.68	37.14	49.28	56.99
#4	✓	✓	✗	✗	✓	40.87	49.09	56.24	63.53	67.29
#5	✓	✓	✓	✗	✓	42.60	49.86	56.87	63.76	67.60
#6	✓	✓	✗	✓	✗	43.79	49.67	56.73	63.49	67.36
#7	✓	✓	✓	✓	✗	44.56	50.26	57.30	64.32	68.19

Table 7 TAPG compare between AAM with attention(Malinowski et al., 2018; Vaswani et al., 2017)

Attention	THUMOS-14					ActivityNet−1.3		
	@50	@100	@200	@500	@1000	AR @100	AUC (val)	AUC (test)
Hard Malinowski et al. (2018)	43.74	49.24	56.63	63.46	67.25	77.11	69.02	69.56
Soft Vaswani et al. (2017)	42.60	49.86	56.87	63.76	67.60	76.93	69.06	69.23
AAM	**44.56**	**50.26**	**57.30**	**64.32**	**68.19**	**77.67**	**69.71**	**70.10**

actors and objects in understanding human action. Comparisons between Exps.(#4 vs. #6) and (#5 vs. #7) highlight the strong impact of AAM.

In Exp.#1 and Exp.#3, as Environment Beholder is not presented, AAM consequently cannot be applied because it requires environment feature as one of its input. Therefore, we replace AAM by a simple soft self-attention layer followed by an average pooling operation to fuse multiple actors together. Likewise, in Exp.#4 and Exp.#5 we also perform the above replacement strategy to emphasize the effectiveness of AAM.

4.5.2 Effectiveness of AAM

We continue studying the effectiveness of the proposed AAM in TAPG task on both ActivityNet−1.3 and THUMOS-14 by comparing AAM with different attention mechanisms, i.e., soft self-attention (Vaswani et al., 2017) (Soft), hard attention (Malinowski et al., 2018) (Hard) as shown in Table.7.

For soft self-attention mechanism, we simply remove the actors hard attention part at the beginning of our AAM, which is defined in Eqs. (2–6), and directly feed the input set of actors features $\mathcal{F}^a$ (or objects features $\mathcal{F}^o$) into a self-attention mechanism.

In contrast, for hard-attention mechanism, we replace the self-attention part at the end of our AAM by a simple average pooling operation to average the selected actors features $\tilde{F}^a$ (or selected objects features $\tilde{F}^o$) into a single representation f^a (or f^o).

With the higher performances on both datasets shown in Table 7, AAM proves its appealing advantages over soft self-attention and hard attention mechanisms.

Table 8 TAPG comparison between our AOE-Net with BMN (Lin et al., 2019) on egocentric videos (Damen et al., 2021)

	AR@10	AR@100	AUC
BMNLin et al. (2019)	11.59	34.26	25.14
AOE-Net	**15.99**	**37.40**	**29.20**

4.5.3 Performance of AOE-Net with Different Number of Objects

The number of input objects K for objects beholder (Sect. 3.1.3) is also a hyper-parameter that may affect the performance of our AOE-Net. If we use a large K, the overall model may receive more noisy information due to the increasingly incorrect detected objects. Whereas, if we use a small K, the objects beholder will not present enough significant information with a few objects. Thus, the contribution of objects beholder to understand actions is insufficient.

In this ablation study, we benchmark our AOE-Net on various number of objects K with TAPG task and ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015) dataset. The comparison is reported in Table 9.

Table 9 shows that as we increase K, the TAPG performance of AOE-Net also tend to improve. However, when $K > 20$, the performance starts fluctuating and is not robust due to more wrongly detected objects in each snippet. Therefore, we conclude that $K = 20$ gives the best trade-off between performance and robustness.

Table 9 TAPG performance of our AOE-Net on ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015) with various settings of K

Number of Objects (K)	AR@100	AUC (val)	AUC (test)
0	77.02	68.98	69.72
1	77.15	69.17	69.95
5	77.45	69.43	69.96
10	77.24	69.26	69.56
20	**77.67**	*69.71*	*70.10*
30	77.55	69.63	69.96
40	**77.67**	**69.86**	**70.22**
50	77.24	69.17	69.81

4.5.4 Robustness of AOE-Net to Egocentric Videos

To benchmark the robustness of AOE-Net on egocentric videos, we use EPIC-KITCHENS 100 (Damen et al., 2021) to benchmark TAPG task. Table 8 provides the TAPG comparison between our AOE-Net with BMN (Lin et al., 2019). Even actors are not shown in the egocentric videos, our AOE still obtains good TAPG performance with a big improvement compare to BMN. This proves the effectiveness of the objects beholder.

4.5.5 Network Efficiency

Table 10 reports the efficiency of AOE-Net and previous SOTA with #parameters in millions (M), computational cost (GFLOPs), inference time on a 3-minute video with either an Intel Core i9-9920X CPU or a single NVIDIA RTX 2080 Ti.

4.6 Qualitative Analysis of AAM

4.6.1 Qualitative Results of AAM with Actors Beholder

Fig. 8 shows qualitative performances of AAM in selecting main actors in the set of detected ones. The videos are retrieved from ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015). In the case of multiple actors detected (Fig. 8a and Fig. 8c, our proposed AAM can effectively select main actors in the scene and remove the insignificant actors. This aims to eliminate redundant information as well as select the most relevant information to feed into the boundary-matching module. Figure 8b illustrates the scenario where the environment is tedious and may not contribute to perceive the action. However, the local information at the bounding box around the main actor can help highlight the action. In Fig. 8b, AAM one again shows its merit when selecting the main actor who actually commits the action.

4.6.2 Performance of AAM Affected by Human Detector

The human detector we used is Faster-RCNN (Ren et al., 2015) trained on COCO dataset (Lin et al., 2014). In practice, the human detector is not completely perfect in the videos due to motion blurs or low resolutions. Therefore, the AAM is also affected by the quality of detected human bounding boxes.

In Fig. 9, we visualize frames of 4 videos where the human detector poorly produces human bounding boxes. In Fig. 9a, the green bounding box is localized around two athletes in the pool beside two separate bounding boxes for each athlete. Although the green bounding box is incorrect because it contains two humans (even three if we count the one behind), it is intuitively better than the individual boxes of each athlete because it contains richer information of the scene. This proves that our AAM effectively learnt to select the best bounding boxes detected regardless of its quality in terms of human detection.

In Fig. 9b, c and d, we notice that there are badly detected bounding boxes which is just a body part of the humans instead of a whole. However, AAM could learn to eliminate these bad bounding boxes to only select the correct ones.

From the above observations, we can see that AAM can learn to avoid selecting bad bounding boxes which do not fully contain the humans. Oppositely, AAM also learns to select some badly detected bounding boxes but contains more meaningful information than the correct ones.

However, we conclude that relying on the human detector in providing locations to attend on is preventing AAM to

Table 10 Network efficiencies of AOE-Net and several of previous works

	#Params (M)	FLOPs (G)	Inference time (s)	
			GPU	CPU
BMN Lin et al. (2019)	4.9	71.22	0.128	4.15
DBG Lin et al. (2020)	2.9	47.52	0.03	-
GTAD Xu et al. (2020)	5.6	150.28	0.14	-
ABN Vo et al. (2021)	6.9	87.88	0.07	0.21
AEI Vo et al. (2021)	6.9	90.62	0.08	0.21
AOE-Net	8.8	94.02	0.12	0.27

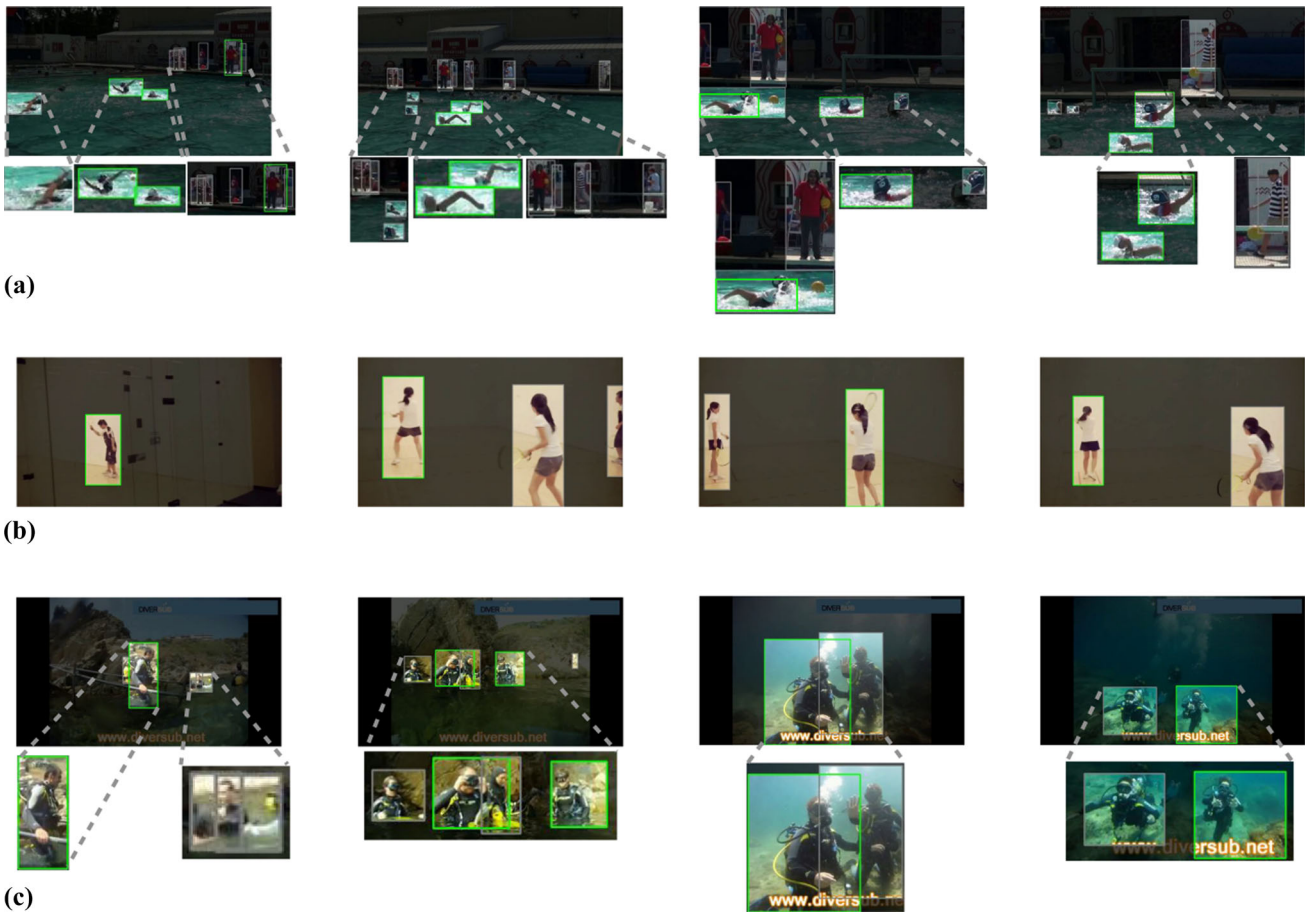


Fig. 8 Visualization of main actors selection resulting by AAM on ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015). **a**, **b**, and **c** are three different videos. The background is blacked out, the bounding boxes of main actors are outlined by green line and the bounding boxes of insignificant actors are outlined by grey line (Color figure online)

achieve its highest potential. Therefore, in the future, it would be more beneficial if we have a better module to localize interesting spatial locations in the video frames instead of the human detector.

4.6.3 Qualitative Analysis of Objects Beholder

Figure 10 visualizes how AOE-Net can take advantage of Objects Beholder. In this figure, we showcase two video for two distinct categories of (A) visible actors and (B) non-visible actors. in (A), the actors are visible and commits the action of tightrope walking, therefore, our AOE-Net can take advantage of all the beholders. Whereas in (B), the actors are not visible in the video frame and commits the action of cooking, therefore, our AOE-Net can only relies on Objects Beholder.

In each illustration of (A) and (B), we visualize either when the action does not happen (A.i and B.i) and when the action is happening (A.ii and B.ii).

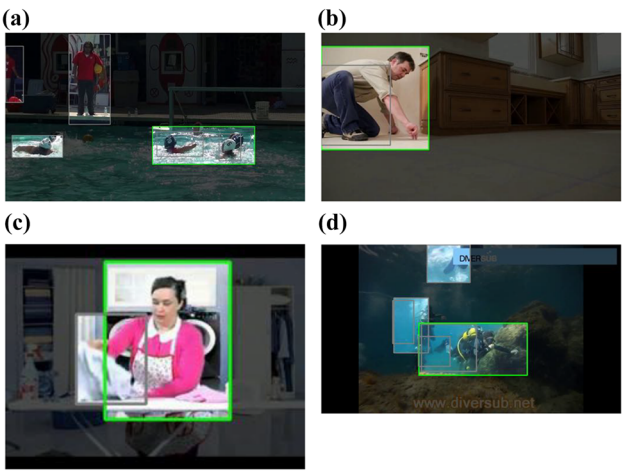


Fig. 9 Visualization of AAM on ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015) on cases where the human detector poorly generates human bounding boxes. The background is blacked out, the bounding boxes of main actors are outlined by green line and the bounding boxes of insignificant actors are outlined by grey line (Color figure online)

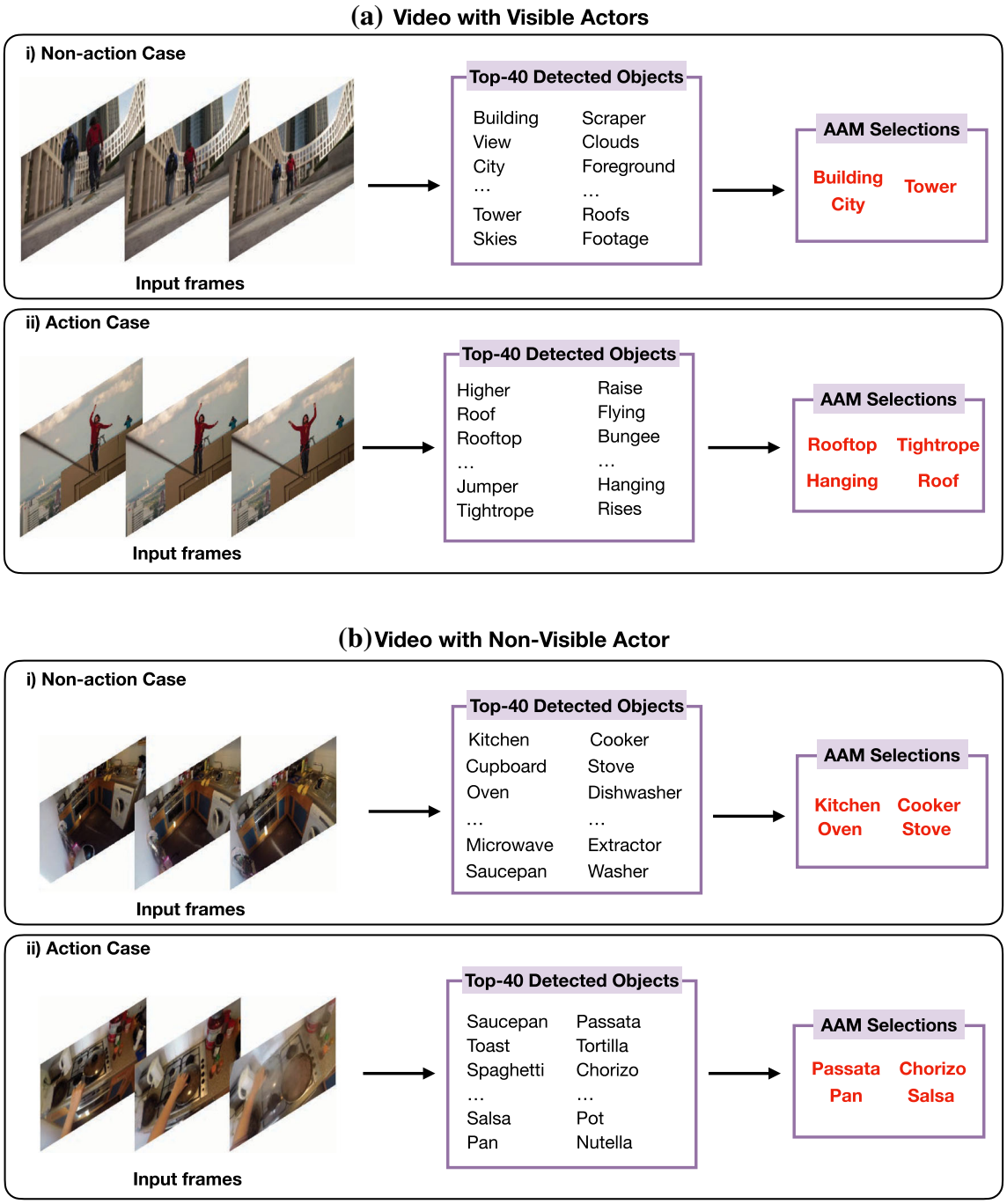


Fig. 10 Qualitative results to illustrate the effectiveness of Objects Beholder with AAM in (**A**) videos with visible actors and (**B**) videos with non-visible actor. In each case, we illustrate two instances of having action and without action. The input frames are shown on the left, its objects detected by CLIP are shown in the middle, and the most relevant objects selected by AAM are shown in the right

As we can see in Fig. 10A, the objects detected in non-action case and action case are very different. Specifically in Fig. 10(A.i), the non-action scene is captured through objects like "City", "Building", "Tower", etc. Whereas in Fig. 10(A.ii), the action scene includes "Rooftop", "Tightrope", "Roof", and "Hanging", etc.

Likewise, in Fig. 10B, the objects detected in each cases of non-action and action are very different. On one hand, in Fig. 10(B.i), detected objects are "Kitchen", "Cooker", "Oven", and "Stove" etc. On the other hand, in Fig. 10(B.ii), the action scene objects consist of "Passata", "Chorizo", "Pan", and "Salsa", etc.

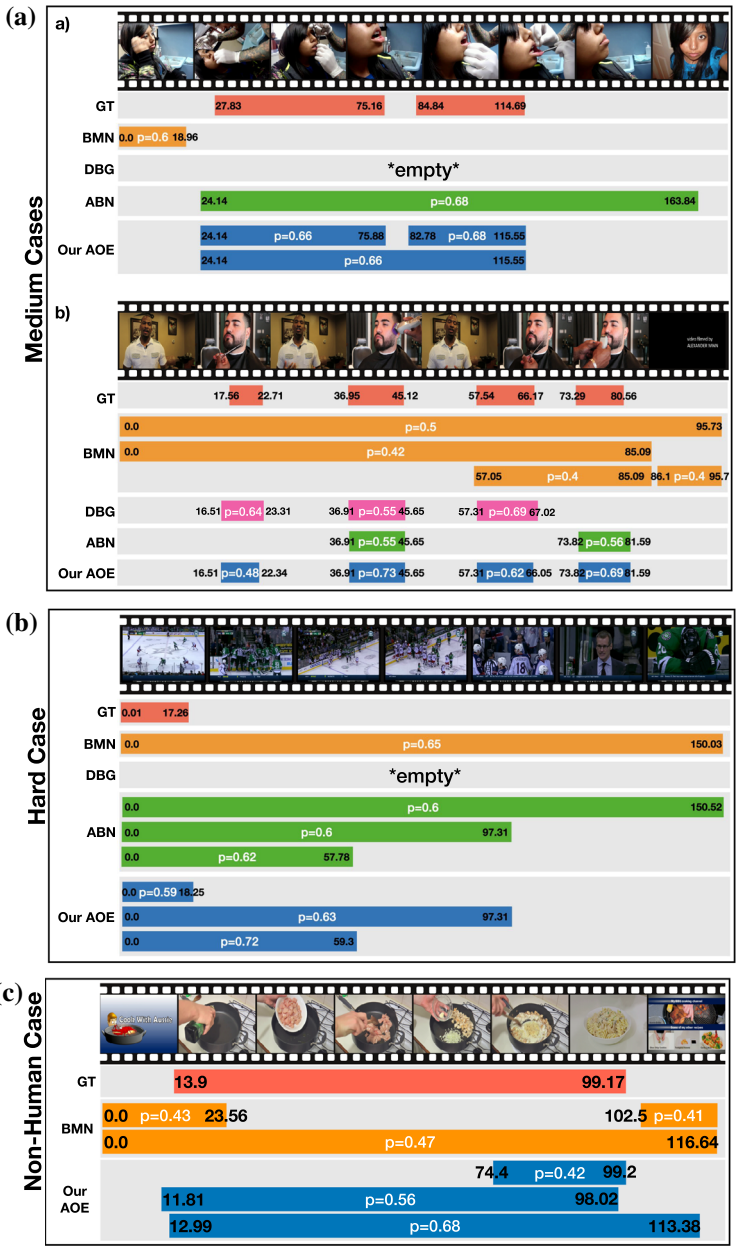


Fig. 11 Qualitative results in TAPG on ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015) dataset

4.7 Qualitative Analysis of AOE-Net

The qualitative results of our AOE-Net in TAPG of ActivityNet−1.3 (Fabian Caba Heilbron and Niebles, 2015) comparing with previous SOTA works (Lin et al., 2019, 2020; Vo et al., 2021) are illustrated in Fig. 11. In this figure, we showed two medium cases and a hard case and case of egocentric video. In each video, the proposals from all methods with higher scores than 0.4 are selected to show in the qualitative examples.

In videos of simple cases (Fig. 11A), the actors who are receiving pierces (or trimming in video(b)) is easily observed from the whole frame, however, the piercing action is a bit difficult to be identified because it's from the hand of the doctor (or the taylor in video (b)), who is outside of the video view. Therefore, BMN (Lin et al., 2019) and DBG (Lin et al., 2020) completely failed to propose the exact action intervals. Likewise, ABN (Vo et al., 2021) is tricked by the video and proposes an interval from the beginning of the first groundtruth action of piercing until the credit cut. On the other hand, our proposed AOE-Net can propose intervals that match with the groundtruth actions. This explains a lot for the contributions of both actors beholder and objects beholder, which provide more informative features than previous works to give good results.

In the video of hard case(Fig. 11A), the actors are hockey players, who appear very small in the video frames. Therefore, the "hockey playing" activity, which appear at the beginning of this video, is very difficult to be distinguished to "celebrating" activity, which takes place right after the former. This is explanable because we need to carefully observe the movements of hockey players carefully to see this difference. Therfore, all BMN and DBG failed to recognize the groundtruth action interval. Meanwhile, ABN can propose an interval that covers the scene of the field but not necessesarily the groundtruth action. On the other hand, our proposed AOE-Net can propose an interval that closely matches the groundtruth action. This again, explains the contributions of our actors and objects beholders.

In the non-human case (Fig. 11C), an actor shows their hands doing cooking on a pan in the interval of [13.9−99.17], while in [0.0−13.9] and [99.17−116.64] the advertisements are displayed. As the actor only shows their hands in the video frames to commit the action, they cannot be detected by the Actors Beholder. However, thanks to our Objects Beholder, the advertisement intervals at the beginning and the end of the video are easily perceived, hence the true action interval is detected in between. Contrarily, a previous SOTA model, BMN [3], mis-perceived the advertisement intervals as true actions and classifies them as separate action intervals, or mistakenly combines them with the true action interval in between.

5 Conclusion and Discussion

In this paper, we attempt to simulate the human perceiving ability and proposed a novel AOE-Net to locate actions in untrimmed videos. Our AOE-Net contains two modules: PMR and BMM. PMR extracts visual-linguistic representation of each snippet with four beholders. Environment beholder and actors beholder capture global and local visual features of environment and main actors, respectively. Objects beholder extracts linguistic feature from relevant objects. The last beholder aims to model the relations between main actors, relevant objects and environment. To focus on an arbitrary number of main actor(s) or relevant objects, we introduced AAM. The qualitative and quantitative results conducted on ActivityNet−1.3 and THUMOS-14 datasets on both TAPG and TAD tasks evidently suggest that our proposed AOE-Net outperforms SOTA methods. To prove the effectiveness of AOE-Net, we provided ablation studies to show the contribution of each beholder, the effectiveness of proposed AAM, network efficiency, as well as the robustness of AOE-Net when performing on egocentric videos with EPIC-KITCHENS 100 dataset. We further investigate the performance of AOE-Net with various backbone network configurations. These results prove that replicating human perceiving ability in video understanding is a promising track to follow and further explore in the future.

There are several potential future directions from this research. First, while main actors and relevant objects provide important impact to both TAPG and TAD tasks, it would be of great interest to investigate the impact of human body parts (e.g. hands, legs) and the interaction between them with objects in localizing human activities in untrimmed videos. Finally, integrating our method with human tracking (i.e. main actors tracking) might result in even better performance.

Acknowledgements This material is based upon work supported by the National Science Foundation (NSF) under Award No OIA-1946391, NSF 1920920, and NSF 2223793. Minh-Triet Tran was funded by Vingroup and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2019.DA19.

Appendix: Notations

The following is a table that summarizes and briefly describes important symbols that are used throughout this manuscript:

Table 11 Descriptions of symbols used in our paper

Symbol	Description
$\mathcal{V}$	A sequence of all frames from the input video
N	Total number of frames in the input video
δ	Length of a snippet, a sub-set of consecutive frames
T	Total number of snippets from the input video
s_i	The snippet at index $i \in T$
I	The frame in the center of snippet s_i
$\mathcal{B}$	A set of human bounding boxes detected in center frame I of snippet s_i
$\phi(.)$	An encoding function to encode a snippet s_i into a feature vector
$\mathcal{F}^{\mathcal{M}}$	A feature map extracted by a backbone network
$\mathcal{F}^a$	A set of actors features in snippet s_i
$\mathcal{F}^o$	A set of objects features in snippet s_i
$\mathcal{T}$	Vocabulary used in Objects Beholder
D	Total number of words in vocabulary $\mathcal{T}$
$\mathcal{T}^e$	A set of embedded features for every word in $mathcalT$
I^e	Embedded feature representing center frame I of s_i
K	A hyper-parameter, defines maximum number of words in $\mathcal{T}$ to be selected
$\mathcal{F}^o$	Top K embedded features in $\mathcal{T}^e$ that is best matched with I
$\hat{f}^e$	Encoded feature of f^e, used in AAM
$\hat{\mathcal{F}}^a$	Encoded feature of every actors feature in $\mathcal{F}^a$, used in AAM
H^a	A set of scores of every actor feature in $\mathcal{F}^a$
τ	An adaptive threshold to filter out actor features that has lower scores
$\tilde{\mathcal{F}}^a$	A set of main actor features that are selected
f^e	Output feature vector of Environment Beholder
f^a	Output feature vector of Actors Beholder
f^o	Output feature vector of Objects Beholder
$\mathcal{F}^{aoe}$	A stack of f^a, f^o, f^e to serve Actors-Objects-Environment Beholder
f^i	Output feature vector of Actors-Objects-Environment Beholder
$\mathcal{L}_s$	Loss function to optimize for starting points classification
L_s	Labels of starting points in the input
$\mathcal{L}_e$	Loss function to optimize for ending points classification
L_e	Labels of ending points in the input video
$\mathcal{L}_{act}$	Loss function to optimize for the actions classification and regression
L_D	Labels for the actions classification and regression
$\mathcal{L}_{wb}$	Weighted binary cross-entropy loss
$\mathcal{L}_2$	Mean squared error loss

References

Lin, T., Zhao, X., Su, H., Wang, C., & Yang, M. (2018). Bsn: Boundary sensitive network for temporal action proposal generation. In ECCV.

Su, H., Gan, W., Wu, W., Yan, J., & Qiao, Y. (2020). BSN++: complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. In ACCV.

Lin, T., Liu, X., Li, X., Ding, E., & Wen, S. (2019). Bmn: Boundary-matching network for temporal action proposal generation. In ICCV.

Lin, C., Li, J., Wang, Y., Tai, Y., Luo, D., Cui, Z., Wang, C., Li, J., Huang, F., & Ji, R. (2020). Fast learning of temporal action proposal via dense boundary generator. AAAI, 11499–11506.

Xu, M., Zhao, C., Rojas, D. S., Thabet, A., & Ghanem, B. (2020). G-tad: Sub-graph localization for temporal action detection. In CVPR.

Vo, K., Yamazaki, K., Truong, S., Tran, M.-T., Sugimoto, A., & Le, N. (2021). Abn: Agent-aware boundary networks for temporal action proposal generation. *IEEE Access, 9,* 126431–126445.

Vo-Ho, V.-K., Le, N., Kamazaki, K., Sugimoto, A., & Tran, M.-T. (2021). Agent-environment network for temporal action proposal generation. In ICASSP, pp. 2160–2164.

Shou, Z., Wang, D., & Chang, S.-F. (2016). Temporal action localization in untrimmed videos via multi-stage cnns. In CVPR.

Gao, J., Yang, Z., & Nevatia, R. (2017). Cascaded boundary regression for temporal action detection. arXiv e-prints, 1705–01180. arXiv:1705.01180 [cs.CV].

Gao, J., Chen, K., & Nevatia, R. (2018). Ctap: Complementary temporal action proposal generation. In ECCV.

Gao, J., Ge, R., Chen, K., & Nevatia, R. (2018). Motion-appearance co-memory networks for video question answering. In CVPR.

Fabian Caba Heilbron, B. G. Victor Escorcia, & Niebles, J. C. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In CVPR, pp. 961–970.

Jiang, Y.-G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., & Sukthankar, R. (2014). THUMOS Challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/.

Krishna, R., Hata, K., Ren, F., Fei-Fei, L., & Carlos Niebles, J. (2017). Dense-captioning events in videos. In ICCV, pp. 706–715.

Kay, W., Carreira, J., , et al. (2017). The kinetics human action video dataset. arXiv preprint arXiv:1705.06950.

Richard, A., & Gall, J. (2016). Temporal action detection using a statistical language model. In CVPR.

Chao, Y., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J., & Sukthankar, R. (2018). Rethinking the faster r-cnn architecture for temporal action localization. In CVPR, pp. 1130–1139.

Heilbron, F. C., Niebles, J. C., & Ghanem, B. (2016). Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In CVPR.

Gao, J., Yang, Z., Chen, K., Sun, C., & Nevatia, R. (2017). Turn tap: Temporal unit regression network for temporal action proposals. In ICCV.

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE TPAMI, 35*(1), 221–231.

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

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In NIPS. NIPS'14, pp. 568–576. MIT Press, Cambridge, MA, USA.

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In ICCV.

Mei, T., Zhang, W., & Yao, T. (2020). Vision and language: from visual perception to content creation. APSIPA TSIP *9*.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In CVPR, pp. 6077–6086

Radford, A., Kim, J. W., et al. (2021). Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020.

Heilbron, F. C., Niebles, J. C., & Ghanem, B. (2016). Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In CVPR, pp. 1914–1923.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In NeurIPS, pp. 91–99.

Lin, T., Goyal, P., Girshick, R., He, K., & Dollör, P. (2017). Focal loss for dense object detection. In ICCV, pp. 2999–3007

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In ICCV, pp. 4489–4497.

Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., & Lin, D. (2017). Temporal action detection with structured segment networks. In ICCV.

Liu, Y., Ma, L., Zhang, Y., Liu, W., & Chang, S.-F. (2019). Multi-granularity generator for temporal action proposal. In CVPR.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI, 20*(11), 1254–1259.

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *TIST, 12*(5), 1–32.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Cho, K., Courville, A., & Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia, 17*(11), 1875–1886.

Galassi, A., Lippi, M., & Torroni, P. (2020). Attention in natural language processing. IEEE Transactions on Neural Networks and Learning Systems.

Chaudhari, S., Polatkan, G., Ramanath, R., & Mithal, V. (2019). An attentive survey of attention models. arXiv preprint arXiv:1904.02874.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. U., & Polosukhin, I. (2017). Attention is all you need. In NeurIPS. Curran Associates, Inc.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In ICML, pp. 2048–2057. PMLR.

Elsayed, G., Kornblith, S., & Le, Q. V. (2019). Saccader: Improving accuracy of hard attention models for vision. In NeurIPS.

Patro, B., & Namboodiri, V. P. (2018). Differential attention for visual question answering. In CVPR, pp. 7680–7688.

Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., & Mei, T. (2019). Gaussian temporal awareness networks for action localization. In CVPR, pp. 344–353.

Liu, S., Zhao, X., Su, H., & Hu, Z. (2020). Tsi: Temporal scale invariant network for action proposal generation. In ACCV.

Bai, Y., Wang, Y., Tong, Y., Yang, Y., Liu, Q., & Liu, J. (2020). Boundary content graph neural network for temporal action proposal generation. In ECCV, pp. 121–137. Springer.

Tan, J., Tang, J., Wang, L., & Wu, G. (2021). Relaxed transformer decoders for direct action proposal generation. ICCV.

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask r-cnn. In ICCV.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1715–1725. Association for Computational Linguistics, Berlin, Germany. https://doi.org/10.18653/v1/P16-1162. https://aclanthology.org/P16-1162.

Dosovitskiy, A., Beyer, L., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. CVPR.

Malinowski, M., Doersch, C., Santoro, A., & Battaglia, P. (2018). Learning visual question answering by bootstrapping hard attention. In ECCV, pp. 3–20.

Vo, K., Joo, H., Yamazaki, K., Truong, S., Kitani, K., Tran, M.-T., & Le, N. (2021). Aei: Actors-environment interaction with adaptive attention for temporal action proposals generation. In 32nd British Machine Vision Conference 2021, BMVC 2021, Virtual Event, UK, November 22-25, 2021. https://www.bmvc2021-virtualconference.com/assets/papers/1095.pdf.

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-nms – improving object detection with one line of code. In ICCV.

Neubeck, A., & Van Gool, L. (2006). Efficient non-maximum suppression. In ICPR, vol. 3, pp. 850–855.

Damen, D., Doughty, H., et al. (2021). Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. IJVC, 1–23.

Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., & Tian, Q. (2020). Bottom-up temporal action localization with mutual regularization. In ECCV, pp. 539–555. Springer.

Dai, X., Singh, B., Zhang, G., Davis, L. S., & Qiu Chen, Y. (2017) Temporal context network for activity localization in videos. In ICCV.

Yao, T., Li, Y., Qiu, Z., Long, F., Pan, Y., Li, D., & Mei, T. (2017). Msr asia msm at activitynet challenge 2017: Trimmed action recognition, temporal action proposals and densecaptioning events in videos. In CVPR Workshops.

Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., & Niebles, J. C. (2017). End-to-end, single-stream temporal action detection in untrimmed videos. In BMVC

Eun, H., Lee, S., Moon, J., Park, J., Jung, C., & Kim, C. (2019). Srg: Snippet relatedness-based temporal action proposal generator. IEEE Transactions on Circuits and Systems for Video Technology, p. 1.

Wang, X., Zhang, S., Qing, Z., Shao, Y., Gao, C., & Sang, N. (2021). Self-supervised learning for semi-supervised temporal action proposal. In CVPR, pp. 1905–1914.

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

Zheng, J., Chen, D., & Hu, H. (2021). Boundary adjusted network based on cosine similarity for temporal action proposal generation. Neural Processing Letters, 1–16.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., & Zitnick, L. (2014). Microsoft coco: Common objects in context. In ECCV.

Gao, J., Shi, Z., Wang, G., Li, J., Yuan, Y., Ge, S., & Zhou, X. (2020). Accurate temporal action proposal generation with relation-aware pyramid network. In AAAI, vol. 34, pp. 10810–10817.

Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., & Gan, C. (2019). Graph convolutional networks for temporal action localization. In ICCV, pp. 7094–7103.

Xiong, Y., Wang, L., Wang, Z., Zhang, B., Song, H., Li, W., Lin, D., Qiao, Y., Gool, L. V., & Tang, X. (2016). CUHK & ETHZ & SIAT submission to activitynet challenge 2016. CoRR arXiv:1608.00797.

Wang, L., Xiong, Y., Lin, D., & Van Gool, L. (2017). Untrimmednets for weakly supervised action recognition and detection. In CVPR.