RCL: Recurrent Continuous Localization for Temporal Action Detection

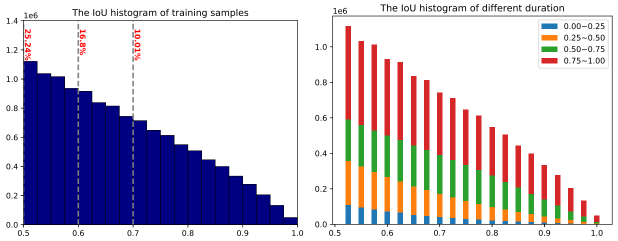
Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan
DAMO Academy, Alibaba Group
{qishi.wq, yanhao.zyh, zhengyun.zy, panpan.pp}@alibaba-inc.com

Abstract

Temporal representation is the cornerstone of modern action detection techniques. State-of-the-art methods mostly rely on a dense anchoring scheme, where anchors are sampled uniformly over the temporal domain with a discretized grid, and then regress the accurate boundaries. In this paper, we revisit this foundational stage and introduce Recurrent Continuous Localization (RCL), which learns a fully continuous anchoring representation. Specifically, the proposed representation builds upon an explicit model conditioned with video embeddings and temporal coordinates, which ensure the capability of detecting segments with arbitrary length. To optimize the continuous representation, we develop an effective scale-invariant sampling strategy and recurrently refine the prediction in subsequent iterations. Our continuous anchoring scheme is fully differentiable, allowing to be seamlessly integrated into existing detectors, e.g., BMN [20] and G-TAD [41]. Extensive experiments on two benchmarks demonstrate that our continuous representation steadily surpasses other discretized counterparts by $\sim$2% *mAP. As a result, RCL achieves* 52.92% *mAP@0.5 on THUMOS14 and* 37.65% *mAP on ActivtiyNet v1.3, outperforming all existing single-model detectors.*

1. Introduction

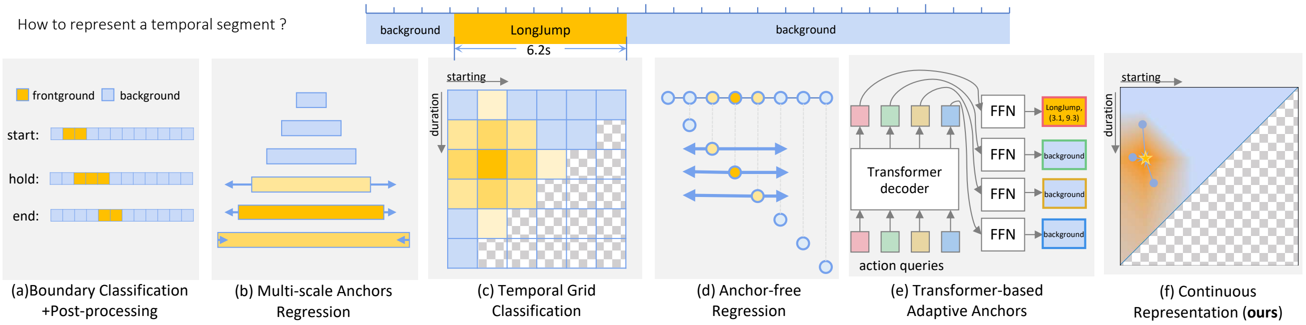
Temporal Action Localization (TAL) that localizes temporal boundaries of actions with specific categories in untrimmed videos [6, 14, 15], is at the core of several downstream tasks such as video classification [12], video captioning [44] and video editing [13]. This challenging problem has been deeply studied in recent years [20, 21, 32, 40, 41], as the large scale variation problems is very serious, causing sophisticated feature designs to capture both local and global information, and thus inspired many extensions such as UNet-like architecture [22], local context [28], and proposal-relations [41, 42, 45]. Prior works [9, 40] take inspiration from image detection [29, 30] and are carried out by densely spanning temporal anchors and predicting their corresponding scores. Other challenges include the fact that the definition of an action's temporal boundaries are often ambiguous [2]. The ambiguity and uncertainty also hinder the convergence for localization optimization, and brings an illogical empirical observation that the classification-based detectors [20, 41] usually achieve better performance than regression-based methods [9].



(a) the overall anchor distribution (b) the unbalanced scale distribution

Figure 1. The tIoU histogram of training anchors from BMN [20]. The red numbers are the positive percentage higher than the corresponding tIoU threshold. These anchors mainly cover the long segments, which results in a missing detection for short instances.

While numerous efforts have been made towards solving the above challenges, recent approaches still suffer from a major limitation: they mainly leverage a *discretized* anchoring representation. For example, existing bottom-up methods [21, 32, 46] utilize the discretized boundary classification and a well-tuned post-processing to compose temporal segments, which can not be trained in an end-to-end manner. Recently, many works utilize the predefined temporal anchor to represent the temporal hypothesis, *e.g.*, the sliding-windows paradigm [33] and the multi-scale anchors [9, 40]. These methods show excellent performance with faster speed and have the ability to handle large duration segments. In contrast to representing complete segments, some anchor-free methods, *e.g.*, AFSD [19], leverage center-point representations to directly regress the start and end time, and the latest studies [35, 39] utilize transformer decoder to bi-match the segments with action queries. In general, different representation methods usually steer the detectors to perform well in different aspects. For example, the bottom-up representation is usually more accurate for fine-grained localization. The anchor-based



(a)Boundary Classification +Post-processing

(b) Multi-scale Anchors Regression

(c) Temporal Grid Classification

(d) Anchor-free Regression

(e) Transformer-based Adaptive Anchors

(f) Continuous Representation (**ours**)

Figure 2. The typical temporal representation methods. (a) the bottom-up representation [21, 46]. (b) the multi-scale anchor representation [9, 40]. (c) the grid-based representation [20, 43]. (d) the anchor-free representation [19]. (e) the transformer-based representation [35] (f) the proposed continuous representation. Best viewed in color.

representation achieves better completeness and is easy to optimize with the tIoU supervision. The anchor-free representations avoid the need for an anchoring design and are usually quite efficient. The transformer has shown powerful abilities with set matching loss from action queries. Noticing that different representations and their anchoring optimization are usually heterogeneous, but their performances essentially depend on the anchor *distribution* and the *ranking* quality between the anchors. As shown in Figure 1, the discretized anchoring representation [20] can only provide coarse proposals, causing seriously missed detection for short-term segments.

To address this issue, we introduce a novel anchoring representation that is efficient, expressive, and fully continuous, as depicted in Figure 2(f). Our key idea is to directly regress confidence scores from continuous anchor points using deep neural networks. Thus we can extract precise segments by searching local maximum in the continuous function.

In this work, we present Recurrent Continuous Localization (RCL), an explicit model conditioned with video embeddings and temporal coordinates. Our approach uses the concept of a Continuous Anchoring Representation (CAR) to achieve high fidelity action detection. Unlike common anchor-based detection techniques, which discretize the segments into a regular grid for measurement [20], we produce an estimation in the continuous field. The proposed continuous representation can be intuitively understood as a learned position-conditioned classifier for which the confidence scores are jointly determined by the video features and the temporal coordinates.

The proposed RCL can serve as a generic plug-in module into various prevalent temporal action localization frameworks, including BMN [20] and G-TAD [41]. Extensive experiments on the THUMOS14 [15] and ActivityNet v1.3 [6] show that RCL substantially improves various detectors by $2 \sim 4\%$ mAP. In particular, we improve a strong BMN detector by about 1.8% average mAP and 5.9% recall, reaching 37.65% mAP on ActivityNet v1.3.

The innovations of this article are as follows:

- We propose a continuous anchoring representation method, which unifies and extends existing anchor-based detector into a continuous regression problem in 2D coordinates.
- To optimize the continuous representation, we develop an effective scale-invariant sampling strategy, which provide accurate ranking scores for short-term segments.
- With an iterative optimization method, our model adaptively focus on target region and provide a refined estimation.
- Our model obtain state-of-the-art results in quantitative comparisons on the THUMOS14 [15] and ActivityNet v1.3 [6] datasets, with 52.92% mAP@0.5, 37.65% mAP, respectively.

2. Related Work

Temporal Action localization (TAL) aims to find all segments in an untrimmed video with their location described by 2D temporal coordinates. To discriminate action segments from background, intermediate geometric candidates and their corresponding features are required. Here we mainly concentrate on the geometric representations, where typical representations used in TAL are illustrated in Figure 2 and summarized below.

Bottom-up representation. Early TAL frameworks [21, 32, 46] involve evaluating the snippet-level probabilities of three action-indicating phases, *i.e.* starting, continuing, and ending; and obtain temporal boundaries via a intensive post-processing step. They provide an intuitive way to determine a segment by two key moments $(\mathbf{x}_s, \mathbf{x}_e)$. While the heuristic merge operation is usually not fully differentiable, which leads to a inferior performance.

Multi-scale anchor representation. Inspired by anchor-based image detection [29, 30], the first family of anchor-based TAL methods [9,40] typically employ the multi-scale anchor representation and attach an auxiliary boundary regression branch to refine these pre-defined anchors. Geometrically, given pre-defined anchors $(\mathbf{a}_{s,k}, \mathbf{a}_{e,k})$, the network simultaneously predicts the confidence score $\mathbf{s}_k$ and the relative offset $(\triangle\mathbf{a}_{s,k}, \triangle\mathbf{a}_{e,k})$. While the large scale variations in the duration make it challenging to recognize localization boundary [9]. The fixed small set of anchors are also less flexible to cover a complexity distribution. Cascaded localization strategies [22, 28] are usually employed to alleviate this issue.

Grid-based representation. In order to further increase the sampling density, a straightforward solution is to densely enumerate all segments and predict the corresponding confidence scores [5]. [20,41,43] ingeniously express this enumeration structure in discretized 2D grids, and optimize a 2D heatmap through 2D/3D convolution. However, due to the squarely growing compute and memory requirements, current methods are only able to handle low resolutions (256×256 or below). With this coarse discretization, the state-of-the-art grid-based TAL approach, BMN [20], can only obtain 40.2% recall rate for short segments, leading a low-fidelity prediction on ActivityNet v1.3 [6]. For an input video with a snippet-level feature size T_s, the sample space of grid-based representation is at a scale of $\mathcal{O}(T_s^2)$.

Anchor-free representation. To reduce the complexity, some recent frameworks [19] use the center point as a simplified representation and directly regress the boundary location. Geometrically, a center point is described by a 1-D vector (x_c) and the hypothesis sample space is in the scale of $\mathcal{O}(T_s)$, which is much more tractable. The strategy of reducing the sample significantly increases the training difficulty for the regression branch. Therefore, the anchor-free methods usually achieve inferior accuracy compared to anchor-based method.

Transformer representation. More recently, [35,39] introduce the transformer architecture [7] to directly predict all segments in parallel, which take advantage of the query-key mechanism and utilize a small set of learned action queries as implicit adaptive anchor.

Continuous representation. Recent 3D rendering works [26, 27] propose to utilize the continuous signed distance functions to represent 3D shapes and eliminate discretization errors. LIIF [10] extends the continuous representation to image coordinate, which can generate arbitrary super-resolution . Inspired by the above methods, we introduce the continuous representation to the temporal domain.

3. Methodology

RCL Overview. In this section we present RCL, a recurrent continuous localization learning approach. Figure 3 illustrates the overall pipeline of our method. Our method formulates temporal segment as a local maximum in a continuous 2D function $G_{\boldsymbol{\theta}}(\mathbf{F}; \mathbf{x})$. We optimize a deep neural network to represent this function, which simultaneously estimates the confidence scores and relative offsets from the snippet features $\mathbf{F}$ and the 2D temporal coordinate $(\mathbf{x}_s, \mathbf{x}_e)$.

The framework takes an untrimmed video frames $\mathbf{V} \in \mathbb{R}^{3\times T\times H\times W}$ as input and estimates all potential temporal segments $\varphi = \{(\mathbf{x}_{s,n}, \mathbf{x}_{e,n}, c_n)\}_{n=1}^{N}$ that may contain known actions, and these segments can be represented as key points in continuous 2D confidence maps. Our method can be distilled down to three stages: (1) video feature extraction, (2) computing continuous 2D confidence maps, and (3) iterative updates, where all stages are differentiable and composed into an end-to-end trainable architecture.

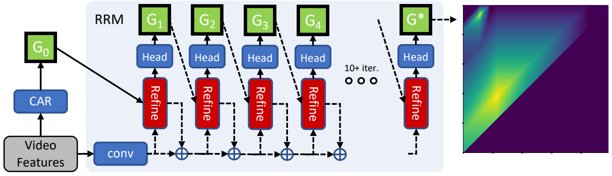


Figure 3. RCL consists of 3 main components: (1) A feature encoder that extracts temporal features from an input video. (2) A continuous anchoring representation (CAR) which predicts a continuous confidence map with a scale-invariant sampling strategy. (3) A recurrent refine module (RRM) which updates the confidence map by iteratively refining the uncertain regions.

3.1. Video Feature Extraction

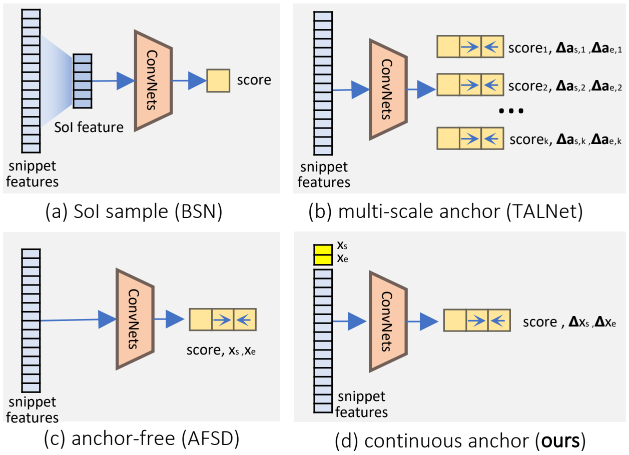
Following the common practice for temporal action detection approaches [20, 21, 41, 42], the video features are offline extracted from the untrimmed video frames using a 3D convolutional network [1,8,12,38]. We adopt the sliding window approach to split the long video into several short snippets, where σ is the time interval and L is the length of a snippet. Our encoder, $\mathbf{F} = f_{\boldsymbol{\theta}}(\mathbf{V})$, utilizes spatial average pooling to eliminate the spatial dimension and outputs a compact video feature $f_{\boldsymbol{\theta}} : \mathbb{R}^{3\times T\times H\times W} \rightarrow \mathbb{R}^{D\times T_s}$, where D is the feature dimension and temporal resolution $T_s = \lfloor (T - L + 1)/\sigma \rfloor$. We use the off-the-shelf video recognition models [1, 38] and freeze the parameters $\boldsymbol{\theta}$ of the video feature extractor $f_{\boldsymbol{\theta}}$ for training efficiency.

3.2. Continuous Anchoring Representation

In this section we present CAR, a continuous representation module, which brings a unified perspective for current geometric representation [9, 19, 21].

As shown in Figure 4, the geometric representation for typical temporal anchoring methods can be formulated as three architectures:

(1) **The bottom-up methods** [21, 46] first obtain the boundary candidates, and then use the 1D RoI pooling (termed "SoI") to estimate all possible combinations. For-



(a) SoI sample (BSN)

(b) multi-scale anchor (TALNet)

(c) anchor-free (AFSD)

(d) continuous anchor (**ours**)

Figure 4. In the continuous representation instantiation, the temporal information and the anchor information are concatenated as input. CAR produces the confidence score and the relative offset for any 2D segment query.

mally, the whole process can be formalized as:

$$\left\{ \begin{aligned} G_{\boldsymbol{\theta}}(\mathbf{F};\mathbf{x}) &= (p_{\boldsymbol{\theta}}(\mathbf{F};\mathbf{x}), \mathbf{x}_s, \mathbf{x}_e) \\ p_{\boldsymbol{\theta}}(\mathbf{F};\mathbf{x}) &= s_{\boldsymbol{\theta}}(\mathbf{F};\mathbf{x}_s) \cdot e_{\boldsymbol{\theta}}(\mathbf{F};\mathbf{x}_e) \cdot q_{\boldsymbol{\theta}}(SoI(\mathbf{F};\mathbf{x}_s,\mathbf{x}_e)), \end{aligned} \right. \tag{1}$$

where $s_{\boldsymbol{\theta}}$, $e_{\boldsymbol{\theta}}$ are two binary classifiers to localize the start time and end time, which are usually implemented with 1D temporal convolution layers. $q_{\boldsymbol{\theta}}$ provides confidence score for a proposal and $p_{\boldsymbol{\theta}}$ is the fused confidence score. BSN [21] adopts the cascaded paradigm to determine the start and end location first, and then composes segments via a boundary-sensitive evaluation. BMN [20] directly enumerates all candidates, and accelerates SoI through a matrix multiplication, which forms an end-to-end training solution. However, since the temporal classifier works on the discretized feature $\mathbf{F}$, the *smallest representable length* is inversely proportional to the feature size $Duration/T_s$ and the size of its sample space is $T_s \cdot (T_s + 1)/2$. To improve the localization accuracy, it is intuitive to rescale the video feature size T_s. While the computational cost will increase significantly, as analyzed in Section 4.3.

(2)**The multi-scale anchor methods** [9, 40] extend image detection, *e.g.* Faster R-CNN [30], to temporal action localization. They generate the class-agnostic proposals by jointly classifying and regressing a fixed set of multi-scale anchors $\mathcal{A} = \{(\mathbf{a}_{s,k}, \mathbf{a}_{e,k})\}_{k=1}^{K}$ at each location. The coordinate transformations are computed as follows:

$$\left\{ \begin{aligned} G_{\boldsymbol{\theta}}(\mathbf{F};\mathbf{a}_k) &= (p_{\boldsymbol{\theta}}(\mathbf{F};\mathbf{a}), \mathbf{a}_{s,k}^*, \mathbf{a}_{e,k}^*) \\ \mathbf{a}_{s,k}^* &= \triangle_{\boldsymbol{\theta}}\mathbf{a}_{s,k} \cdot l_k + \mathbf{a}_{s,k} \\ \mathbf{a}_{e,k}^* &= \triangle_{\boldsymbol{\theta}}\mathbf{a}_{e,k} \cdot l_k + \mathbf{a}_{e,k} \end{aligned} \right. \tag{2}$$

where $\mathbf{a}_k$ and l_k are the coordinate and length for the k-th anchor. Theoretically, the ground-truth segment can be losslessly recovered through the offset regression learning. While the design of the anchor itself is a discretized representation, which will cause an imbalance sample problem [23, 34] and make it less flexible.

(3)**The anchor-free methods** [19] directly predict the confidence score, the center offset and length of time through the center point feature:

$$\left\{ \begin{aligned} G_{\boldsymbol{\theta}}(\mathbf{F};c_i) &= (p_{\boldsymbol{\theta}}(\mathbf{F};c_i), c_i^* - l_{\boldsymbol{\theta},i}/2, c_i^* + l_{\boldsymbol{\theta},i}/2) \\ c_i^* &= \triangle_{\boldsymbol{\theta}} c_i + c_i. \end{aligned} \right. \tag{3}$$

This design makes the system much efficient, but the offset optimization will be more difficult, usually resulting in performance degradation.

(4)**The continuous representation** proposes modeling action segments by maximizing the confidence scores in a 2D function. The key difference to the grid-based methods [20, 41] is that the confidences are defined on a *continuous* temporal domain. For a given segment $(\mathbf{x}_s, \mathbf{x}_e)$, the continuous function can output the segment's confidence score and relative offset to the closest annotation:

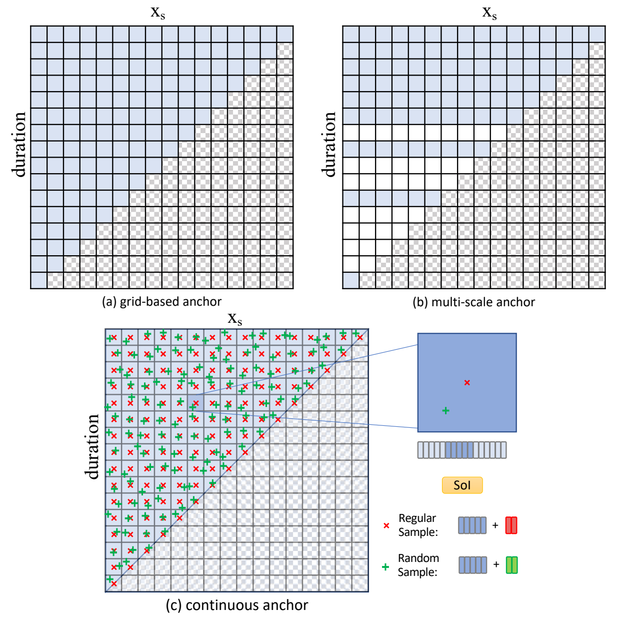
$$\left\{ \begin{aligned} G_{\boldsymbol{\theta}}(\mathbf{F};\mathbf{x}) &= (p_{\boldsymbol{\theta}}([\mathbf{F},\mathbf{x}];\mathbf{x}), \mathbf{x}_s^*, \mathbf{x}_e^*) \\ \mathbf{x}_s^* &= \triangle_{\boldsymbol{\theta}}\mathbf{x}_s \cdot (\mathbf{x}_e - \mathbf{x}_s) + \mathbf{x}_s \\ \mathbf{x}_e^* &= \triangle_{\boldsymbol{\theta}}\mathbf{x}_e \cdot (\mathbf{x}_e - \mathbf{x}_s) + \mathbf{x}_e \end{aligned} \right. \tag{4}$$

where $[,]$ denotes a concatenation operator. Our model is an explicit setting method, which fed the anchor coordinate itself as a *condition* input to constitute the prediction. Note that this design differs essentially from the current anchoring schemes (as Figure 4) in that every location is associated with a dynamic anchor instead of a set of pre-defined anchors. Since it allows arbitrary length, our scheme can better represents the extremely fine-grained segments. Our experiments show that due to the high-fidelity sample space, we achieve much higher recall than the baseline scheme, please refer to Section 4.4.

The continuous design enables more flexible and efficient data sampling space, which shows some appealing properties in Section 3.3.

3.3. Sampling Strategy and Feature Alignment

The proposed representation can be viewed as a continuous extension to the discretized grid representation. In the actual training process, there are two problems: (1) The continuous representation function contains infinite samples, exhaustive sampling is computationally prohibitive. A common solution is to randomly collect some points in each training batch to optimize the overall function [26, 27]. (2) For each ground-truth segment $(\mathbf{g}_s, \mathbf{g}_e)$, it can be mapped to a point on our 2D axis (Figure 5). While prior studies [20, 23] shown that the training samples for different scales are not balanced, the loss terms will be overwhelmed by the long segments. For a continuous representation, we



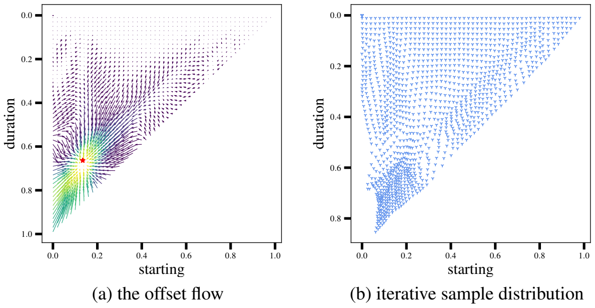
(a) grid-based anchor (b) multi-scale anchor (c) continuous anchor

Figure 5. An illustration of the sample strategies for (a) grid-based anchor representation, (b) multi-scale anchor representation and (c) the proposed continuous anchor representation. The lightblue box denote the anchors for the discretized representation (a-b).

can sample on the entire real number domain, which ensures that we can easily control the ratio for different length instances.

To solve the above issues, we propose a scale-invariant sampling strategy. (1) **Regular Grid Samples**: Note that when only sample points in the regular grid centers (× in Figure. 5 (c)), our continuous representation can degenerate to grid representation [20] (Figure 5 (a)). The valid samples number from 2D-grid is $T_s \cdot (T_s + 1)/2$. (2) **Random Samples**: To train the continuous function, we random sample $T_s \cdot (T_s + 1)/2$ segments around the regular grid samples (+ in Figure 5 (c)). (3) **Scale-invariant Samples**: For each ground-truth annotation $(\mathbf{g}_{s,i}, \mathbf{g}_{e,i})$, we sample n points around it, taking its length l as the variance. As shown in Figure 1, for a long segment, there may be hundreds of samples. In this case, there are relatively more pairs for long segments. The number of short-term samples is rare, and the learning of ranks between samples is relatively difficult. Our balanced sample strategy is very helpful for the rank learning between instances of different lengths.

In addition, we note that although our method, as a black-box function, can predict a confidence score for arbitrary segment. However, according to Shannon's sampling theorem [31], our finest input observation is the video frame, and the temporal resolution of our output is still limited. Therefore, we keep the minimum output duration at the video frame level, termed SPF (Seconds per Frame).



(a) the offset flow (b) iterative sample distribution

Figure 6. An illustration of the offset flow (a), which is predicted from regular grid, and the updated sample distribution (b).

3.4. Recurrent Refine Module

As shown in Figure 3, the update operator takes base video feature $\mathbf{F}$, confidence maps $\mathbf{G}_m$, and a latent hidden state as input, and outputs updated confidence maps $\mathbf{G}_{m+1}$ and an updated hidden state. With each iteration, it produces an update direction $(\triangle \mathbf{x}_s, \triangle \mathbf{x}_e)$, and then we perform lookups on the continuous 2D grid (Figure 6. These steps are repeated until convergence. The architecture of our update operator is designed to mimic the steps of the progressive boundary refinement [22, 28]. The update operator is trained to perform refinement such that the sequence converges to a fixed state $\mathbf{G}_m \rightarrow \mathbf{G}^*$.

The iterative prediction architecture, following [36], refines the predictions over successive stages, $m \in \{1, ..., M\}$, with intermediate supervision at each stage. More details are in the supplementary materials. Section 4.3 analyzes the accuracy and generalization for this module.

3.5. Supervision

Given a set of ground-truth segment annotations $\mathcal{G} = \{\mathbf{g}_n = (\mathbf{g}_{s,n}, \mathbf{g}_{e,n})\}_{n=1}^{N}$, the current anchor-based approaches [20, 41] heavily rely on tIoU scores as the supervisory signal:

$$\begin{cases} \mathcal{L}_{tIoU} = \mathcal{L}_{bce}(p^1_{\boldsymbol{\theta}}, \mathbf{1}\{\text{tIoU}^* > \tau\}) + \lambda_1 \mathcal{L}_{mse}(p^2_{\boldsymbol{\theta}}, \text{tIoU}^*) \\ \text{tIoU}^*(\mathbf{x}, \mathcal{G}) = \max\limits_{\mathbf{g}_n \in \mathcal{G}}(\{|\mathbf{x} \cap \mathbf{g}_n| \,/\, |\mathbf{x} \cup \mathbf{g}_n|\}), \end{cases} \tag{5}$$

where τ is the front-ground threshold, $p^1_{\boldsymbol{\theta}}$ and $p^2_{\boldsymbol{\theta}}$ are two type of confidence maps, $\mathcal{L}_{bce}$ is a balanced cross entropy loss, $\mathcal{L}_{mse}$ is the mean square loss. We argue that the tIoU score is actually a non-signed distance [27]. When each training sample is optimized independently, the network cannot perceive the accurate target location, which leads to a slow convergence [25].

Therefore, we add a signed regressing loss as auxiliary supervision signals to predict the time offset for each segment, which shows a better overall performance (see Table 4) . We adopt the original confidence losses [20] with a boundary regression loss $\mathcal{L}_{offset}$ as below:

$$\begin{cases} \mathcal{L}_{reg} = \mathcal{L}_{tIoU} + \lambda_2 \mathcal{L}_{offset} \\ \mathcal{L}_{offset} = |\triangle \mathbf{x} - (\mathbf{g}^* - \mathbf{x})|, \end{cases} \tag{6}$$

where $\mathbf{g}^*$ denotes the closest ground-truth annotation to the input segment $\mathbf{x}$.

For fair comparisons with our baselines [20, 41], we retain the boundary regularization (TEM Loss in BMN [20] and Node Classification Loss in GTAD [41]) and the ℓ-2 parameter regularization loss:

$$\mathcal{L}_{norm} = \mathcal{L}_{boundary} + \lambda_3 \mathcal{L}_{\ell\text{-}2}(\boldsymbol{\theta}). \tag{7}$$

For the iterative process (Section 3.4), we use the same loss function, but the truncated return training method is used, and α is given as the attenuation parameter. The intermediate supervision at each stage addresses the vanishing gradient problem by replenishing the gradient periodically [36]. The overall objective is

$$\mathcal{L}_{all} = \sum_{m=1}^{M} \alpha^m \mathcal{L}_{reg,m} + \mathcal{L}_{norm}. \tag{8}$$

4. Experiments

In this section, we firstly introduce two standard datasets, THUMOS14 [15] and ActivityNet v1.3 [6], to evaluate the localization ability and the configuration details of our algorithm. Meanwhile, we compare the proposed method, RCL, with existing representative approaches on the two benchmarks. Then we carry out the ablation experiments to explore the contribution of each component in our method. Finally, we further explore the results on various attributes.

4.1. Datasets and Evaluation Metrics

Datasets and features. We validate our proposed method on two standard datasets: THUMOS14 [15] includes 413 untrimmed videos with 20 action classes. According to the public split, 200 of them are used for training, and 213 are used for testing. There are more than 15 action annotations in each video; ActivityNet v1.3 [6] is a large-scale temporal action localization dataset with 200 classes annotated. The entire 19,994 untrimmed videos are divided into training, validation, and testing sets by ratio 2:1:1. Each video has around 1.5 action instances. To make a fair comparison with the previous works, we use the same two-stream features of these datasets. The two-stream features, which are provided by [38], are extracted by I3D network [8] pre-trained on Kinetics [16]. We further validate the effectiveness of our approach with a strong pre-trained feature TSP [1].

Implementation details. We reimplemented BMN [20] and G-TAD [41] following their respective papers as two discretized baselines. We follow the original papers' training schedules and train our model end-to-end using Adam [17] with batch size of 16. The learning rate is 6×10^{-6} on THUMOS14 and 1×10^{-3} on ActivityNet v1.3 for the first 5 epochs, and is reduced by 10 for the following 5 epochs. During training, we set weighting parameter $\lambda_1 = 10, \lambda_2 = 1, \lambda_3 = 10^{-5}, \alpha = 0.8$, the front-ground threshold $\tau = 0.7$ and training iteration $M = 10$. During inference, following [41], we take the segments classification scores from the tIoU and classification branch, and multiply them to produce the proposal score and then fuse our prediction scores with video-level classification scores from [37, 38]. For post-processing, we apply Soft-NMS [4], where the threshold is 0.3 and select the top-Q prediction for final evaluation, where Q is 100 for ActivityNet v1.3 and 200 for THUMOS14.

Table 1. **Temporal Action detection results on test set of THUMOS14**, measured by mAP (%) at different tIoU thresholds. Our RCL achieves the highest mAP for tIoU threshold 0.5 (commonly adopted criteria), significantly outperforming all other methods.

Method	0.3	0.4	0.5	0.6	0.7	Short
End-to-end learned/finetuned on THUMOS for TAL						
TCN [11]	-	33.3	25.6	15.9	9.0	-
R-C3D [40]	44.8	35.6	28.9	-	-	-
PBRNet [22]	58.5	54.6	51.3	41.8	29.5	-
Pre-extracted features						
TAL-Net [9]	53.2	48.5	42.8	33.8	20.8	-
P-GCN [42]	63.6	57.8	49.1	-	-	-
I.C&I.C [46]	53.9	50.7	45.4	38.0	28.5	49.1
MGG [24]	53.9	46.8	37.4	29.5	21.3	-
BSN [21]	53.5	45.0	36.9	28.4	20.0	-
DBG [18]	57.8	49.4	39.8	30.2	21.7	-
BMN [20]	56.0	47.4	38.8	29.7	20.5	-
G-TAD [41]	54.5	47.6	40.2	30.8	23.4	44.2
BC-GNN [3]	57.1	49.1	40.4	31.2	23.1	-
PBRNet* [22]	54.8	49.2	42.3	33.1	23.0	43.6
VSGN [45]	66.7	60.4	52.4	41.0	30.4	56.6
RCL (ours)	**70.1**	**62.3**	**52.9**	**42.7**	**30.7**	**57.1**

* Results are referred from [45]. They replace 3D convolutions with 1D convolutions to adapt to the feature dimension.

Metric for temporal action localization. To evaluate the performance for TAL, we use mean Average Precision (mAP) metric. On THUMOS14 dataset, we report the mAP with multiple tIoUs in set $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. As for ActivityNet v1.3 dataset, the tIoU set is $\{0.5, 0.7, 0.95\}$. Moreover, we also report the averaged mAP where the tIoU is from 0.5 to 0.95 with a stride of 0.05.

4.2. Comparisons with State-of-the-Arts

We compare the proposed RCL with recent state-of-the-art methods on the THUMOS14 dataset. As shown in Table 1, with the same pre-trained features, RCL significantly surpasses the grid baseline [20] by absolute 14.1% mAP@0.5, reaching 52.9% mAP@0.5 on THUMOS14. RCL also demonstrates competitive performance with top-

Table 2. **Action localization results on the validation set of ActivityNet v1.3**, measured by mAPs at different tIoU thresholds and the average mAP. Our RCL, without further finetuning, achieves the state-of-the-art average mAP for most pre-extracted features.

Method	0.5	0.75	0.95	Average	Short
End-to-end learned/finetuned on ActivityNet for TAL					
CDC [32]	45.30	26.00	0.20	23.80	-
R-C3D [40]	26.80	-	-	-	-
PBRNet [22]	53.96	34.97	8.98	35.01	-
Pre-extracted I3D [8] features					
TAL-Net [9]	38.23	18.30	1.30	20.22	-
P-GCN [42]	48.26	33.16	3.27	31.11	-
I.C & I.C [46]	43.47	33.91	**9.21**	30.12	14.8
PBRNet* [22]	51.32	33.33	7.09	33.08	17.6
VSGN [45]	**52.32**	35.23	8.29	**34.68**	**18.8**
RCL (ours)	51.74	**35.27**	8.03	34.39	18.5
Pre-extracted TSN [38] features					
BSN [21]	46.45	29.96	8.02	30.03	15.0
BMN [20]	50.07	34.78	8.29	33.85	15.2
G-TAD [41]	50.36	34.60	9.02	34.09	17.5
BC-GNN [3]	50.56	34.75	**9.37**	34.26	-
PBRNet* [22]	51.41	34.35	8.66	33.90	18.0
VSGN [45]	52.38	36.01	8.37	35.07	19.9
TCANet [28]	52.27	**36.73**	6.86	35.52	-
RCL (ours)	**54.19**	36.19	9.17	**35.98**	**20.0**
Pre-extracted TSP [1] features					
G-TAD [41]	51.26	37.12	**9.29**	35.81	19.3
VSGN [45]	53.26	36.76	8.12	35.94	20.9
RCL (ours)	**55.15**	**39.02**	8.27	**37.65**	**21.1**

* Results are referred from [45]. They replace 3D convolutions with 1D convolutions to adapt to the feature dimension.

Table 3. **Effectiveness of RCL components on the validation set of ActivityNet v1.3.** CAR is highly effective for short actions. RRM improve the overall performance.

Baseline	CAR	RRM	0.5	0.75	0.95	Avg.	Short
✓			50.07	34.78	8.02	33.85	17.5
✓	✓		52.22	36.45	7.53	35.41	18.6
✓	✓	✓	**54.19**	**36.19**	**9.17**	**35.98**	**20.0**

Table 4. **Ablation study for the continuous representation on the validation set of ActivityNet v1.3.** A naive rescaling leads to a slight decrease in accuracy.

input dim	output dim	AP@0.5	mAP	FLOPs (G)
100	100×100	50.07	33.85	45.6
200	200×200	51.56	33.54	91.2
300	300×300	51.60	33.95	136.8
100	200×200	50.79	33.11	45.6
100	300×300	50.60	33.01	45.6
100	CAR w/o offset	**52.35**	34.81	63.2
100	CAR w/ offset	52.22	**35.41**	98.3

Table 5. **Ablation study for the sample strategies on the validation set of ActivityNet v1.3**. Our sampling performs better than regular grid sampling and scale-invariant loss [23] indicating the importance of continuous sampling.

sample strategy	AP@0.5	mAP	FLOPs (G)
regular grid sample	50.07	33.85	45.6
+uniform sample	51.60	34.14	58.8
+scale-invariant sample	**52.35**	**34.81**	63.2
scale-invariant loss [23]	51.18	34.15	45.6

Table 6. **Ablation study for recurrent refine module on the validation set of ActivityNet v1.3.**

Update	backbone	AP@0.5	mAP
	TSN	52.35	35.41
✓	TSN	**54.19** **(+1.84)**	**35.98** **(+0.57)**
	TSP	53.76	36.33
✓	TSP	**55.15** **(+1.39)**	**37.65** **(+0.32)**

performing temporal action method [45], which leverages strong data augmentation and an innovative graph network. Compared with other iterative optimization method [22], our algorithm has substantially denser samples, which brings 10.6% mAP@0.5 improvement.

Table 2 shows the TAL performace with different features [1, 8, 38] on ActivityNet v1.3. Among the compared detectors using TSN features, RCL provides the best results with an mAP of 35.98%. RCL achieves a substantial improvement over BMN, with a gain of 1.56% average mAP via TSN features. Among the compared detectors with TSP [1] features, RCL achieves new state-of-the-art performances with mAP scores of 37.65%. Comparing our RCL with the discretized counterpart [41], it shows a remarkable gains with 1.8% average mAP, indicating the effectiveness of our detection network under challenging fine-grained scenarios.

4.3. Ablation Study

We evaluate the key components of Continuous Anchoring Representation (CAR) and the Recurrent Refine Module (RRM) with TSN features. From Table 3, we can see CAR obviously improves the performance of short actions as well as the overall performance with +1.56% average mAP. We apply the recurrent module to the refine the heatmaps, which performs 0.57% better than not using the recurrent module. This shows that the recurrent optimization indeed helps the model to find the right segments because the initial prediction is a very rough estimation with less context.

To further reveal the devil in the details, a set of simple designs are collected:

The continuous representation: We first design two naive structures to improve the scale: (1) directly scaling the input size and (2) using bilinear layer to up-sample the final heatmaps. As shown in Table 4, we find that mAP is reduced from 33.85% to 33.54% and 33.01%, respectively. We compare these two upsampling structures to our learned representation and find that the continuous representation cascaded regression significantly help promote the mAP.

Sample strategies: In Table 5, we argue that the proposed scale-invariant sample strategy can mitigate the imbalanced

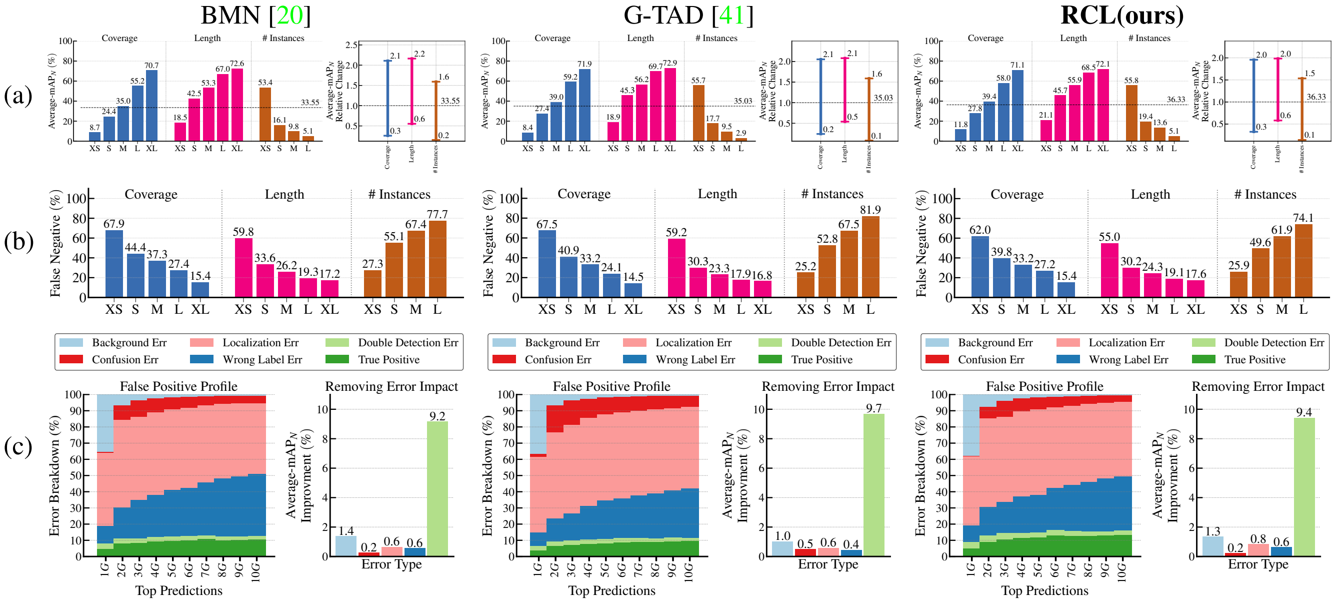


Figure 7. Illustration of the three types of DETAD analyses [2] in ActivityNet v1.3 [6]. (a) The sensitivity average-mAP$_N$ to action characteristics shows RCL mainly benefits from identifying the tiny segments. (b) The false positive profiles shows RCL significantly reduces the missing detection by ∼5.5% for "Extremely Small" instances. (c) Average false positive profile across algorithms for each characteristic. Actions are divided into five duration groups (seconds): XS (0, 30], S (30, 60], M (60, 120], L (120, 180], and XL (180, inf). Please refer to [2] for more details.

data distribution. Moreover, instead of directly increasing the weight for a small number of short-term samples, our dense sample strategy can eliminate overfitting and provide a stable estimation for ranking temporal proposals.

Recurrent refine module: Table 6 shows that the incremental refinements consistently outperform the accuracy on all features. As a supplement to the offset regression branch, we solve the boundary refinement in an iterative way.

4.4. DETAD [2] Error Analysis

To demonstrate the potential gaps with the discretized counterparts, BMN [20] and G-TAD [41] and analyze the sensitivity, we show comparisons over three types of DETAD analyses [2] on ActivityNet v1.3 [6] with TSP features [1]. Figure 7 provides meaningful insights for how continuous representation improve the overall performance. In Figure 7(a), we can see that the mAP$_N$ is reduced from 72.1% to 21.1% with different segment lengths. The sharp decline shows that the low detection accuracy of tiny (XS/S) instances is an important bottleneck restricting the overall performance. Our RCL consistently outperforms the two baseline methods on tiny instances: Coverage-XS (+3.1/+3.4%), Coverage-S (+3.4/+0.4%), Length-XS (+2.6/+2.2%), Length-S (+3.2/+0.4%). This result shows that employing continuous representation is helpful for learning fine-grained clips and thus improves performance.

In addition, Figure 7(b) reveals that RCL achieves the lowest false negative rate with Coverage-XS (-5.9/-5.5%), Coverage-S (-4.6/-1.1%), Length-XS (-4.8/-4.2%), Length-S (-3.4/-0.1%). The superior performance on false negative profile clearly demonstrates that RCL mitigates the resolution issue of tiny instance and allows to represent shorter segments than other detectors.

Finally, we conduct false positive profile to verify the limitations for our detector and show results in Figure 7(c). The most impact error comes from double detection error, which may suffer from the inherent problem in Soft-NMS [4] with low tIoU threshold. We hope that a totally end-to-end continuous representation will be a future work.

5. Conclusion

In this paper, we propose a continuous representation, which brings a unified perspective for current anchoring representation. The proposed representation builds upon an explicit model conditioned with video embeddings and temporal coordinates, which can generate non-uniform anchors of arbitrary length. We develop an effective scale-invariant sampling strategy and recurrently refine the prediction in subsequent iterations. The experimental results on the THUMOS14 and ActivityNet v1.3 datasets show the notable performance gain over current state-of-the-art methods, demonstrating that our RCL can detect high fidelity segments. We hope RCL can serve as a simple yet effective baseline for the community.

References

[1] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3173–3183, 2021. 3, 6, 7, 8

[2] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 256–272, 2018. 1, 8

[3] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 6, 7

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 6, 8

[5] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017. 3

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 1, 2, 3, 6, 8

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 3, 6, 7

[9] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018. 1, 2, 3, 4, 6, 7

[10] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 3

[11] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5793–5802, 2017. 6

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 1, 3

[13] Nathan Frey, Peggy Chi, Weilong Yang, and Irfan Essa. Automatic non-linear video editing transfer. *CoRR*, abs/2105.06988, 2021. 1

[14] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2782–2795, 2013. 1

[15] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014. 1, 2, 6

[16] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 6

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11499–11506, 2020. 6

[19] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2021. 1, 2, 3, 4

[20] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[21] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 2, 3, 4, 6, 7

[22] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11612–11619, 2020. 1, 3, 5, 6, 7

[23] Shuming Liu, Xu Zhao, Haisheng Su, and Zhilan Hu. Tsi: Temporal scale invariant network for action proposal generation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 4, 7

[24] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3604–3613, 2019. 6

[25] Yi Liu, Limin Wang, Xiao Ma, Yali Wang, and Yu Qiao. Fineaction: A fined video dataset for temporal action localization. *arXiv preprint arXiv:2105.11107*, 2021. 5

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 3, 4

[27] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 3, 4, 5

[28] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 485–494, 2021. 1, 3, 5, 7

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 2

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1, 2, 4

[31] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001. 5

[32] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5734–5743, 2017. 1, 2, 7

[33] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1058, 2016. 1

[34] Haisheng Su, Weihao Gan, Wei Wu, Yu Qiao, and Junjie Yan. Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. *arXiv preprint arXiv:2009.07641*, 2020. 4

[35] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13526–13535, October 2021. 1, 2, 3

[36] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 5, 6

[37] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4325–4334, 2017. 6

[38] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018. 3, 6, 7

[39] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. *arXiv preprint arXiv:2106.11149*, 2021. 1, 3

[40] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017. 1, 2, 3, 4, 6, 7

[41] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[42] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019. 1, 3, 6, 7

[43] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12870–12877, 2020. 2, 3

[44] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13278–13288, 2020. 1

[45] Chen Zhao, Ali K Thabet, and Bernard Ghanem. Video self-stitching graph network for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13658–13667, 2021. 1, 6, 7

[46] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *European Conference on Computer Vision*, pages 539–555. Springer, 2020. 1, 2, 3, 6, 7