Abstract—Recent temporal action proposal generation approaches have suggested integrating segment- and snippet score-based methodologies to produce proposals with high recall and accurate boundaries. In this paper, different from such a hybrid strategy, we focus on the potential of the snippet score-based approach. Specifically, we propose a new snippet score-based method, named Snippet Relatedness-based Generator (SRG), with a novel concept of “snippet relatedness”. Snippet relatedness represents which snippets are related to a specific action instance. To effectively learn this snippet relatedness, we present “pyramid non-local operations” for locally and globally capturing long-range dependencies among snippets. By employing these components, SRG first produces a 2D relatedness score map that enables the generation of various temporal intervals reliably covering most action instances with high overlap. Then, SRG evaluates the action confidence scores of these temporal intervals and refines their boundaries to obtain temporal action proposals. On THUMOS-14 and ActivityNet-1.3 datasets, SRG outperforms and refines their boundaries to obtain temporal action proposals. Furthermore, compared to competing proposal generators, SRG leads to significant improvements in temporal action detection.

Index Terms—Temporal action proposal generation, temporal action detection, snippet relatedness, pyramid non-local block, SRG.

I. INTRODUCTION

PEOPLE can easily share and watch large numbers of videos due to the growth of digital media platforms. The analysis of these real-world videos has attracted considerable attention in the computer vision community. One of the major tasks is temporal action detection, which aims to detect action instances in a long untrimmed video. Specifically, the detection process includes localizing temporal boundaries and classifying the class of each instance. Temporal action detection can be used in many applications such as smart surveillance [1], [2], video summarization [3], [4], [5], and video retrieval [6], [7], [8].

Similar to object proposals for object detection [9], [10], temporal action proposals play an important role in temporal action detection [12], [13], [14]. Specifically, many methods [15], [16], [17], [18] for temporal action detection perform proposal generation followed by action classification. For this reason, the performance of temporal action detection can be improved by generating a few proposals that capture temporal action instances with high overlap.

Temporal action proposal generation methods [12], [14], [18], [19], [20], [21], [22], [23] can be categorized into segment-based, snippet score-based, and hybrid methods. Segment-based methods [12], [14], [19], [20] directly use sliding windows to define temporal segments and evaluate their action confidence scores. By adopting a large number of segments with several temporal spans, these methods achieve high recall performance. However, uniformly distributed segments...
provide inaccurate boundaries for action instances.

To address the inaccurate boundary problem of segment-based proposals, snippet score-based methods [22], [18] have been proposed, where a snippet is defined as a set of a few consecutive frames [12]. In such methods, actionness or boundary scores are evaluated for individual snippets, and the locations of snippets with high scores are then grouped to form proposals. Although snippet-level evaluation can provide accurate boundaries, low-quality snippet scores often result in missing action instances.

Hybrid methods [23], [21] aim to generate proposals with both high recall and accurate boundaries by combining segment- and snippet score-based methods. These methods achieve higher performance than other types of methods. However, hybrid methods still exploit the low-quality 1D score sequences for fusion processes.

To overcome the problems mentioned above, in this paper, we present a novel concept of “snippet relatedness” to consider the local relations among snippets. The snippet relatedness is related to the local consistency of consecutive frames, which plays an important role in tasks requiring the relations among frames or images [24], [25], [26]. Specifically, we propose a new snippet score-based approach, named Snippet Relatedness-based Generator (SRG), with the snippet relatedness (see Fig. 1). Snippet relatedness indicates which snippets are related to a specific action instance. Based on the snippet relatedness, we evaluate relatedness scores that indicate whether neighboring snippets \( s_{nbr} = (s_{nbr}^-, s_{nbr}^+) \) belong to the same action instance as a reference snippet \( s_{ref} \) (see the middle of Fig. 1). Here, \( s_{nbr}^- \) and \( s_{nbr}^+ \) denote the past and future snippets of \( s_{ref} \), respectively. Note that our relatedness score is different from the actionness score.

First, a relatedness score sequence considers only the action instance to which the reference snippet belongs, while an actionness score sequence includes all action instances in an input video. Second, existing snippet score-based methods directly use a 1D actionness score sequence to obtain proposals. Unlike this, we produce a 2D relatedness score map by manipulating the multiple 1D score sequences for all snippets. To effectively learn the snippet relatedness, we present “pyramid non-local (PN) operations” that locally and globally capture long-range dependencies among snippets. By employing these components, SRG first produces three types: segment-based, snippet score-based, and hybrid methods. Segment-based methods directly use sliding windows for temporal segments and evaluate their action confidence scores. S-CNN-prop [14] adopts C3D [38] as a binary classifier to evaluate confidence scores of segments.

• Different from actionness scores, we manipulate multiple 1D score sequences to produce a 2D relatedness score map that enables the generation of various temporal intervals covering most action instances with high overlap.

• We present “pyramid non-local (PN) operations” to effectively learn the snippet relatedness by locally and globally capturing long-range dependencies.

• We conduct extensive experiments on large-scale datasets, where SRG achieves state-of-the-art performance for temporal action proposal generation.

The rest of this paper is organized as follows. In Section II, we review previous works on action recognition, temporal action detection, and temporal action proposal generation. In Section III, the details of the proposed method are delineated in two subsections. Experiments on benchmark datasets are described to verify the effectiveness and superiority of our approach in Section IV. In Section V, we discuss the limitations of the proposed method. Finally, we conclude this paper in Section VI.

II. RELATED WORKS

A. Action Recognition

Action recognition on a trimmed video has been studied extensively in the past. Earlier works [29], [30], [31] exploited hand-crafted features. Recently, convolutional neural networks (CNNs) have driven impressive progress. Among the various architectures of CNNs, two-stream networks [32], [33], [34], [35], [36] have achieved state-of-the-art performance by either separately or jointly learning appearance and motion features. 3D CNNs [27], [38] have also achieved excellent performance by learning spatio-temporal features.

B. Temporal Action Detection

Temporal action detection predicts the action categories, start times, and end times of action instances in an untrimmed video. S-CNN [14] uses multiple 3D CNNs consisting of proposal, classification, and localization networks for temporal action detection. R-C3D [39] and TAL-Net [15] adopt the framework of Faster R-CNN [11] with a modification for considering large variations in action duration. Shou et al. [40] proposed convolutional-de-convolutional filters that perform spatial downsampling and temporal upsampling operations to learn high-level action semantics and temporal dynamics. Gao et al. [41] employed cascaded boundary regressors to improve the temporal action boundaries. Zhao et al. [18] proposed a structured temporal pyramid and a completeness classifier to produce a global action representation.

C. Temporal Action Proposal Generation

We categorize temporal action proposal generation methods into three types: segment-based, snippet score-based, and hybrid methods. Segment-based methods directly use sliding windows for temporal segments and evaluate their action confidence scores. S-CNN-prop [14] adopts C3D [38] as a binary classifier to evaluate confidence scores of segments.
produced relatedness, starting, and ending score maps. In each score map, each row describes a score sequence with respect to a reference snippet $s_{\text{ref}}$. Fig. 2. Framework of our approach. (a) Given a snippet-level feature sequence of an input video, Temporal Interval Generation Network (TIGN) first generates temporal intervals with high recall and accurate boundaries. To this end, we introduce a CNN model, called Temporal Interval Generation Network (TIGN), which produces three types of 2D score maps (i.e., relatedness, starting, and ending) for an input video.

**Temporal Interval Generation Network (TIGN).** Given $F_S$ as input, TIGN produces relatedness, starting, and ending score maps. In detail, the relatedness scores indicate whether neighboring snippets $s_{\text{nbr}} = (s_{\text{nbr}}^t, s_{\text{nbr}}^+)$. Then, snippet score-based temporal intervals are generated based on the relatedness and weighted relatedness score sequences. (b) By using the temporal intervals and the interval-level features, Temporal Interval Evaluation Network (TIEN) evaluates the action confidence scores of the temporal intervals and refines their boundaries to define temporal action proposals.

**Snippet score-based methods.** Similar to previous methods, we define a snippet sequence $S = \{s_t\}_{t=1}^{L_S}$ of an input video consisting of $N_V$ frames, where $L_S$ is the length of $S$. $S$ is the length of $S$. $L_S$ is calculated as $N_V/I_n$, where $I_n$ is the number of frames in a snippet. By using a snippet $s_t$ as input of the two-stream network, we obtain an appearance feature vector $f^a_t \in \mathbb{R}^{d_a}$ and a motion feature vector $f^m_t \in \mathbb{R}^{d_m}$, where $d_a$ and $d_m$ are the dimensions of each feature vector. Then, a snippet-level feature sequence $F_S$ is denoted as follows:

$$F_S = ((f^a_1, \cdots, f^a_{L_S}), (f^m_1, \cdots, f^m_{L_S})).$$

**Temporal Interval Generation Network (TIGN).** Given $F_S$ as input, TIGN produces relatedness, starting, and ending score maps. In detail, the relatedness scores indicate whether neighboring snippets $s_{\text{nbr}} = (s_{\text{nbr}}^t, s_{\text{nbr}}^+)$ are related snippets. We define related snippets as snippets in $s_{\text{nbr}}$ that belong to the same action instance as $s_{\text{ref}}$. Similar to the relatedness score, the starting (ending) score represents whether the locations of $s_{\text{nbr}}$ ($s_{\text{nbr}}^+$) and $s_{\text{ref}}$ are the starting (ending) of the action instance that $s_{\text{ref}}$ belongs to. For clarity, we present toy examples to illustrate the concepts of related, starting, and ending snippets in Fig. 3. We consider all snippets in $F_S$ as reference snippets, which yields a 1D score sequence for each snippet. Consequently, TIGN generates three types of 2D score maps by manipulating the corresponding types of multiple 1D score sequences. Note that TIGN does not use the sliding window technique to generate these 2D score maps, but produces them by processing a
Fig. 3. Toy examples of three types of snippets. We first define a reference snippet $s_{\text{ref}}$ and its neighboring snippets $s_{\text{nbr}} = (s_{\text{nbr}}^{-}, s_{\text{nbr}}^{+})$. Consecutive snippets in green represent an action instance and each region in sky blue represents the currently considered snippets. (a) Related snippets are the snippets that belong to the same action instance as $s_{\text{ref}}$. (b) The starting snippet is the starting of the action instance that $s_{\text{ref}}$ belongs to. (c) The ending snippet is the ending of the action instance that $s_{\text{ref}}$ belongs to.

snippet sequence for the whole input video at once without any redundant operations.

**TIGN architecture.** As shown in Fig. 4, TIGN consists of three types of blocks, namely, attention, PN, and output blocks. Motivated by recent successful works on attention [43], [44], we introduce two attention blocks to focus on important appearance and motion features, individually. For each attention block, we adopt CBAM [44] and change the spatial attention to temporal attention.

Given the concatenated outputs of each attention block as input, the PN block performs a temporal pyramid pooling operation and the non-local operation [45] (see the top of Fig. 5). In [46], spatial pyramid pooling is successfully used to encode local and global context information. Inspired by this, we introduce a temporal pyramid pooling operation involving average pooling over several temporal spans. Furthermore, we apply the non-local operation to each temporal pooling feature and the residual feature to effectively encode the relatedness among all snippets. Then, we upsample the outputs of the non-local operations to the length of the input snippet sequence. Finally, we concatenate these upsampled features to be fed into a convolutional layer.

In the output block, we employ three convolutional layers and three activation functions to estimate a relatedness score map $O_r$, a starting score map $O_s$, and an ending score map $O_e$ (see the bottom left of Fig. 5). For the three convolutional layers, the numbers of filters $f_r$, $f_s$, and $f_e$ are set to the lengths of $O_r$, $O_s$, and $O_e$, respectively. We use sigmoid activation for $O_r$ and softmax activation for $O_s$ and $O_e$.

**TIGN training.** For training TIGN, we design a multi-task loss function $L^\text{TIGN}$ as follows:

$$L^\text{TIGN} = L^\text{TIGN}_r + L^\text{TIGN}_s + L^\text{TIGN}_e,$$

where $L^\text{TIGN}_r$, $L^\text{TIGN}_s$, and $L^\text{TIGN}_e$ are loss functions defined for the relatedness, starting, and ending score maps, respectively. We adopt a binary cross-entropy loss for $L^\text{TIGN}_r$ and a multi-class cross-entropy loss for $L^\text{TIGN}_s$ and $L^\text{TIGN}_e$ as follows:

$$L^\text{TIGN}_r = -\frac{1}{L_S L_S} \sum_{i=1}^{L_S} \sum_{j=1}^{L_S} \left( M_{r,ij} \log(O_{r,ij}) \right) + \left( 1 - M_{r,ij} \right) \log(1 - O_{r,ij}) \right),$$

$$L^\text{TIGN}_s = -\frac{1}{L_S} \sum_{i=1}^{L_S} \sum_{j=1}^{L_S} \left( M_{s,ij} \log(O_{s,ij}) \right),$$

$$L^\text{TIGN}_e = -\frac{1}{L_S} \sum_{i=1}^{L_S} \sum_{j=1}^{L_S} \left( M_{e,ij} \log(O_{e,ij}) \right).$$
and $O_{e,i}$ does not exist.

![Three Types of Score Label Maps](image)

**Algorithm 1:** Temporal Interval Generation 1

Input: a relatedness score map $O_r$, a threshold value $\tau$

for $i = 1$ to $L_S$
do
  for $j = 1$ to $L_S$
do
    if $O_{r,ij} \geq \tau$
do
      group consecutive temporal locations in $j$
do
      select the location group including $s_{ref,i}$
do
      collect $p_i = (t_{s,i}, t_{e,i})$ of the selected group
  enddo
  add $j$ to $j$
do
enddo

Output: a set of temporal intervals $P$

**Algorithm 2:** Temporal Interval Generation 2

Input: a relatedness score map $O_r$, a starting score map $O_s$, an ending score map $O_e$, a threshold value $\tau$

Initialization: a weighted relatedness score map $\hat{O}_r \leftarrow 0$
do
compute a binary weight map $W$ from $O_s$ and $O_e$
do
for $i = 1$ to $L_S$
do
  for $j = 1$ to $L_S$
do
    if $O_{r,ij} \geq \tau$
do
      group consecutive temporal locations in $j$
do
      select the location group including $s_{ref,i}$
do
      collect $p_i = (t_{s,i}, t_{e,i})$ of the selected group
  enddo
  add $j$ to $j$
do
enddo

Output: a set of temporal intervals $P$

where $M_r \in \mathbb{R}^{L_S \times L_r}$, $M_s \in \mathbb{R}^{L_S \times L_s}$, and $M_e \in \mathbb{R}^{L_S \times L_e}$ are label maps for the relatedness, starting, and ending scores, respectively, and $L_S$, $L_r$, and $L_s$ are the widths of the corresponding label maps. We annotate $M_r$, $M_s$, and $M_e$ based on action instances in a snippet sequence, as shown in Fig. 6. $M_r$ can be annotated as previously described with regard to the concept of the snippet relatedness. For $M_s$ and $M_e$, we add the index “none” for the non-existence of an action instance to enable the use of softmax activation for each row.

**Temporal intervals.** We generate two types of temporal intervals $p = (t_s, t_e)$ based on the score sequences in $O_r$, $O_s$, and $O_e$, as shown in Fig. 7. Here, $t_s$ and $t_e$ are the starting and ending locations, respectively, of the temporal interval. Let $O_{r,i}$, $O_{s,i}$, and $O_{e,i}$ denote the $i$th score sequences in $O_r$, $O_s$, and $O_e$, respectively.

Given $O_{r,i}$, for the first type of temporal interval, we initially obtain the temporal locations of high-score snippets by performing a thresholding operation with a threshold value of $\tau$. We then group the consecutive temporal locations of high-score snippets. Among the location groups, we select only the location group that includes $s_{ref,i}$ to collect $p_i = (t_{s,i}, t_{e,i})$. We apply this process to all relatedness score sequences in $O_r$ to obtain as many reliable temporal intervals as possible, as sketched in Alg. 1.

For the second type of temporal interval, we first obtain the temporal locations of starting and ending snippets with the highest scores in $O_{s,i}$ and $O_{e,i}$, respectively. We then generate a binary weight sequence $W_i$ with values of 0 for locations outside the boundaries and 1 for locations inside the boundaries:

$$W_{ij} = \begin{cases} 1, & \text{if } j_s \leq j \leq j_e, \\ 0, & \text{otherwise} \end{cases},$$

where $j_s$ and $j_e$ are the starting and ending locations of the $i$th corresponding score sequences, respectively. Afterward, we compute a weighted relatedness score sequence $\hat{O}_{r,i}$ by combining the relatedness score sequence $O_{r,i}$ with the boundary-based binary weight sequence $W_i$ as follows:

$$\hat{O}_{r,i} = (O_{r,i} + W_i)/2,$$
where the addition and division are element-wise operations. From \( \hat{O}_{r,i} \), we then collect a temporal interval \( p_i = (t_{s,i}, t_{c,i}) \) by applying the same rule used for the first type of temporal interval generation (see Alg. 2).

By employing both types of temporal intervals, we define a set of temporal intervals \( P = \{p_n\}_{n=1}^{N_P} \), where \( N_P \) is the number of temporal intervals.

**B. Temporal Interval Evaluation**

In this step, we evaluate the action confidence scores for temporal intervals and adjust their boundaries to generate temporal action proposals. To that end, we introduce a CNN model, named Temporal Interval Evaluation Network (TIEN), which uses interval-level features for temporal intervals as input.

**Interval-level features.** We define an interval-level feature \( F_I \) by aggregating snippet-level features within the starting and ending locations of a temporal interval. Similar to \([21],[22]\), we exploit contextual information by considering boundary regions. Specifically, we use snippet-level features within \([t_{s} - L_C], (t_{c} + L_C)\) to form \( F_I \) as follows:

\[
F_I = ((f_{t_{s} - L_C})^a, \ldots, f_{t_{s}}, \ldots, f_{t_{c} + L_C}),
\]

where \( L_C \) is the length of contextual snippets in the boundary region.

**Temporal Interval Evaluation Network (TIEN).** Given a set of interval-level features as input, TIEN performs action confidence evaluation and boundary adjustment. Interval-level evaluation provides more reliable results than snippet-level evaluation due to the consideration of global information. As the output of TIEN, we obtain the action confidence score, starting offset, and ending offset for each temporal interval.

**TIEN architecture.** We construct TIEN using attention, PN, and output blocks (see Fig. 4). The attention and PN blocks are the same as those of TIGN. For the output block, we use a global average pooling layer, a fully connected layer, and sigmoid activation to obtain an action confidence score \( c \).

**TIEN training.** To train TIEN, we use temporal intervals generated by TIGN as training samples. We calculate temporal intersection-over-union (IoU) scores between the temporal intervals and action instances to define action confidence scores \( e^g \) as the ground truth. Among all of the temporal intervals, we use those with \( c^g \geq 0.5 \) as positive samples and those with \( c^g < 0.1 \) as negative samples. The loss function \( L_{\text{TIEN}} \) is defined as follows:

\[
L_{\text{TIEN}} = L_{c}^{\text{TIEN}} + \alpha (L_{s}^{\text{TIEN}} + L_{e}^{\text{TIEN}}),
\]

where \( L_{c}^{\text{TIEN}}, L_{s}^{\text{TIEN}}, \) and \( L_{e}^{\text{TIEN}} \) are loss functions defined for the action confidence scores, starting offsets, and ending offsets, respectively. \( \alpha \) is a parameter for balancing the contributions among the loss functions, which is empirically set to 0.1. We adopt the \( L_1 \) distance for all three loss functions as follows:

\[
L_{c}^{\text{TIEN}} = \frac{1}{N_P} \sum_{n=1}^{N_P} (|c_n - e_n^g|),
\]

\[
L_{s}^{\text{TIEN}} = \frac{1}{N_P} \sum_{n=1}^{N_P} e_n^s (|o_{s,n} - o_n^g|),
\]

\[
L_{e}^{\text{TIEN}} = \frac{1}{N_P} \sum_{n=1}^{N_P} e_n^e (|o_{e,n} - o_n^g|),
\]

where \( e_n^g \) is a binary label that is set to 1 for a positive sample and 0 for a negative sample.

**Temporal action proposals.** By using the output of TIEN, we newly generate a set of temporal action proposals denoted by \( P = \{p_n\}_{n=1}^{N_P} \), where \( p_n = (t_{s,n}, t_{c,n}, o_{s,n}, o_{e,n}, c_n) \). To remove redundant proposals, we additionally perform non-maximum suppression (NMS) in terms of temporal overlap and action confidence scores.

**IV. EXPERIMENTS**

**A. Datasets**

**THUMOS-14.** THUMOS-14 [27] contains 1,010 and 1,574 untrimmed videos in validation and test sets, respectively. Among them, 200 and 212 videos in the validation and test sets, respectively, have temporal annotations with 20 action classes. We trained our models on the validation set and conducted an evaluation on the test set, as done in previous works [12], [15], [21], [22], [23].

**ActivityNet-1.3.** ActivityNet-1.3 [28] consists of 19,994 untrimmed videos containing 200 action classes, where the training, validation, and test sets contain 10,024, 4,926, and 5,044 videos, respectively. Since annotation for the test set is not publicly available, we trained SRG on the training set and conducted an evaluation on the validation set.

**B. Temporal Action Proposal Generation**

**Snippet-level features.** We used the spatial and temporal networks in [46] to extract snippet-level features. Both networks are based on Inception-v3 [47] and were trained on Kinetics-400 [32]. We set the number of frames per snippet to 6 and 18 for THUMOS-14 and ActivityNet-1.3, respectively.

**Implementation details.** For THUMOS-14, we set the numbers of both \( s^-_{nbr} \) and \( s^+_{nbr} \) in TIGN to 600 based on the maximum action length in the training videos. Thus, \( f_r, f_s, \) and \( f_e \) in the output block were set to 1201, 602, and 602, respectively. For the PN block of TIGN, we empirically set the number of the pyramid level to 4 with kernel and stride size pairs of (3, 1), (5, 3), (7, 5), and (15, 7). We used the same number of the pyramid level with kernel and stride size pairs of (1, 3), (3, 3), (5, 3), and (7, 3) for the PN block of TIEN. We trained TIGN for 15 epochs with a learning rate that was initially set to \( 1 \times 10^{-4} \) and exponentially decayed every 10 steps with a base value of 0.96. We trained TIEN for \( 1 \times 10^5 \) steps with the same learning rate strategy used for TIGN. We used batch sizes of 1 and 256 for TIGN and TIEN, respectively. For the interval-level features, we set \( L_C \) to 20.
and rescaled the temporal length of the features to 128. $L_C$ was adopted to consider contextual information, as in previous works \cite{21, 22}. To select the optimal value of $L_C$, we tested values of $L_C$ such as 0, 5, 10 and 20. Among these values, $L_C = 0$ yielded the worst performance with a relatively high loss value, whereas $L_C = 20$ resulted in convergence at a lower loss value compared with $L_C = 5$ and $L_C = 10$.

For ActivityNet-1.3, we set the numbers of both $s_{nbr}^r$ and $s_{nbr}^e$ in TIGN to 540 to consider the maximum action length in the training set. Thus, $f_r$, $f_e$, and $f_c$ in the output block were set to 1,081, 542, and 542, respectively. We used the same kernel and stride sizes used for the PN block on the THUMOS-14 dataset. The batch sizes for training TIGN and TIEN were also the same as the values used for THUMOS-14. We trained TIGN for 10 epochs with a learning rate that was initially set to $1 \times 10^{-4}$ and exponentially decayed every 100 steps with a base value of 0.96. We trained TIEN for $3 \times 10^5$ steps with the same learning rate strategy used for TIGN. We rescaled the temporal length of the interval-level features to 128. We set $L_C$ to 10 because this value empirically yielded the best training results.

For both datasets, we set $\tau$ to 0.1 with an increasing step size of 0.1 to generate temporal intervals based on the relatedness and weighted relatedness score sequences. Note that, unlike TAG \cite{13}, CTAP \cite{21}, and MGG \cite{23} using actionness scores, we do not group multiple temporal intervals since our score sequences consider only the action instance related to the current reference snippet. For NMS, we empirically set the threshold to 0.83 on THUMOS-14 and adaptively set the threshold to $1 - N_P \times 10^{-4}$ on ActivityNet-1.3.

**Evaluation metrics.** As done in previous studies, we computed the average recall (AR) with multiple tIoU thresholds. We used tIoU thresholds of [0.50 : 0.05 : 1.00] and [0.50 : 0.05 : 0.95] for THUMOS-14 and ActivityNet-1.3, respectively. On the THUMOS-14 dataset, we measured the AR by varying the average number of proposals (AN) from 50 to 500. On ActivityNet-1.3, we computed AR@AN=100 and the area under AR vs. AN (AUC). For the AUC, we considered AN values of [1 : 1 : 100].

**Performance comparison.** For the THUMOS-14 dataset, we plot the AR vs. AN and Recall@AN=100 vs. tIoU curves of SRG and 6 previous methods, S-CNN-prop \cite{14}, TAG \cite{18}, TURN \cite{12}, CTAP \cite{21}, BSN \cite{22}, and MGG \cite{23}, in Fig. 8, where SRG outperformed state-of-the-art proposal generation methods on both AR vs. AN and Recall@AN=100 vs. tIoU curves. We further compared SRG to 9 previous methods, DAPs \cite{20}, S-CNN-prop \cite{14}, SST \cite{19}, TURN \cite{12}, TAL-Net-prop \cite{15}, TAG, CTAP, BSN, and MGG, by evaluating the AR with varying AN. As shown in Table I, SRG achieved the best performance among these state-of-the-art methods at every AN.

On ActivityNet-1.3, we compared SRG to four state-of-the-art proposal generators, SSAD-prop \cite{48}, CTAP, BSN, and MGG, in terms of AR@AN=100 and the AUC. As shown in Table II, SRG outperformed the state-of-the-art methods in terms of AR@AN=100 and achieved comparable performance.

![Graph](image-url)
Fig. 9. Qualitative evaluation of our temporal action proposals on THUMOS-14 (first, second, and third rows) and ActivityNet-1.3 (fourth, fifth, and sixth rows). Clear images represent action frames whereas the images covered by semiopaque masks represent background frames. The left and right numbers of a bar indicate the start and end times, respectively.

Qualitative evaluation. In Fig. 9, we present a qualitative evaluation of our temporal action proposals on THUMOS-14 and ActivityNet-1.3. Our proposals reliably capture action instances with accurate boundaries and high action confidence scores.

C. Ablation Study

We conducted comprehensive ablation experiments to investigate the effectiveness of individual components of SRG. We first performed a qualitative evaluation of the three types of score maps (i.e., relatedness, starting, and ending) produced by TIGN. Note that the quality of these score maps is important for the generation of temporal intervals with both high recall and accurate boundaries. We visualize these score maps as obtained on both THUMOS-14 and ActivityNet-1.3 in Fig. 10, where the score maps were reliably evaluated for both short and long action instances.

Second, we measured the recall with varying tIoU values for the temporal intervals obtained from the relatedness and weighted relatedness score sequences. As shown in Table IV, high recall and accurate boundaries were achieved for both types of temporal intervals. We also confirmed that incorporating both types of temporal intervals enables further improvements over the individual performances. These temporal intervals can lead to high-quality proposals with high recall, accurate boundaries, and low false positive rates.
Fig. 10. Visualization of the three types of score maps generated on THUMOS-14 (first and second rows) and ActivityNet-1.3 (third and fourth rows). We visualize the ground truth for the relatedness score map, where the left and right halves are the regions considered for the starting and ending score maps.

**TABLE IV**

| Recall@TIOU for temporal intervals generated from relatedness and weighted relatedness score sequences on THUMOS-14. RS: Relatedness Score Sequence; WRS: Weighted Relatedness Score Sequence. |
|-----|-----|-----|-----|-----|-----|
| RS  | WRS | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| ✓   | ✓   | 93.78 | 90.22 | 83.74 | 71.28 | 49.26 |
| ✓   | ✓   | 93.75 | 89.59 | 83.71 | 70.26 | 46.28 |
| ✓   | ✓   | 95.31 | 92.16 | 87.30 | 75.64 | 53.69 |

To study the effectiveness of TIGN, TIEN, and the PN block, we measured the performance on six cases as follows:

**TIGN<sub>CM</sub>:** In TIGN, we replaced the PN block with a block consisting of two pairs of convolutional and max pooling layers, abbreviated as CM, which is identical to the early part of the PN block (see Fig. 6). We used the temporal intervals generated by this TIGN as proposals. For the action confidence evaluation, we jointly trained actionness scores and then averaged the scores of the snippets in each proposal.

**TIGN<sub>PN</sub>:** We used the temporal intervals generated by TIGN with the PN block as proposals. Same as TIGN<sub>CM</sub>, we jointly trained actionness scores and averaged the scores to obtain the confidence score.

**TIGN<sub>CM</sub> + TIEN<sub>CM</sub>:** We generated temporal intervals by using TIGN with the CM block. To obtain temporal action proposals, we then evaluated the action confidence scores for the temporal intervals and adjusted their boundaries by using TIEN with the CM block.

**TIGN<sub>PN</sub> + TIEN<sub>CM</sub>:** We produced temporal intervals by using TIGN with the PN block. Next, we acquired proposals by using TIEN with the CM block that evaluates the confidence scores for the temporal intervals and refines their boundaries.

**TIGN<sub>CM</sub> + TIEN<sub>PN</sub>:** We used TIGN with the CM block to generate temporal intervals. Then, TIEN with the PN block was used to evaluate the confidence scores for the temporal intervals and refine their boundaries to generate proposals.
TIGN\textsubscript{PN} + TIEN\textsubscript{PN}: We used TIGN with the PN block to generate temporal intervals. We then estimated the confidence scores for the temporal intervals and adjusted their boundaries by using TIEN with the PN block for generating proposals. This is the proposed method, namely SRG.

As shown in Table VII, TIGN\textsubscript{PN} achieved better performance than TIGN\textsubscript{CM}, which verifies the effectiveness of using the PN block in TIGN. Moreover, we achieved a significant performance gain by employing TIEN, which demonstrates that action confidence scores evaluated by TIEN are more reliable than actionness scores. In addition, the higher improvement of TIEN\textsubscript{PN} than TIEN\textsubscript{CM} also proves the effectiveness of using the PN block in TIEN.

We performed an additional experiment to investigate the possibility of boosting performance by directly incorporating snippet-level scores into interval-level scores. The effectiveness of this strategy has been verified in several works [49], [50], [51]. In our work, the relatedness scores of snippets and the action confidence scores of temporal intervals are regarded as snippet-level scores and interval-level scores, respectively. To combine these two types of scores, we first converted the relatedness score map into a snippet-level score sequence by averaging the scores for each temporal location. Then, we obtained a score for a given temporal interval by averaging the snippet-level scores corresponding to the locations within that temporal interval. By multiplying this score by the action confidence score, we acquired a new action confidence score. In Table VII, we compare the performances of SRG with that of the boosted SRG, where the strategy of combining these two different scores successfully improved the performance on THUMOS-14.

### V. Discussion

In this section, we discuss the limitations of the proposed method. In Fig. 11, we show failure cases of relatedness score maps for some short action instances in ActivityNet-1.3. This is because most of the videos in ActivityNet-1.3 contain a single long instance; thus, TIGN may produce less accurate results on short action instances than long action instances. These results can be refined by the boundary refinement of TIEN. We further analyze the ActivityNet-1.3 and THUMOS-14 [27] datasets. Specifically, the videos in the ActivityNet-1.3 training set contain only 1.5 action instances around the center and 36% background per video, on average, while the videos in THUMOS-14 contain an average of 15.8 instances and 71% background per video [15]. These characteristics that often appear in real-world videos make the THUMOS-14 dataset more challenging. We also observed marginal performance increases at a high level of AR on ActivityNet-1.3 and significant improvements at a low level of AR on THUMOS-14 (see Fig. 12).

| Proposal Generator | Classifier | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|-------------------|------------|-----|-----|-----|-----|-----|
| S-CNN-cls [14]    | 36.3       | 28.7 | 19.0 | 10.3 | 5.3 |
| TIGN             | 4.83       | 8.02 |
| TIGN\textsubscript{CM} | 23.0       |
| SRG + [53]       | 46.53      | 46.9 | 39.1 | 31.4 | 22.2 |

**Table VIII** Comparison of temporal action detection performance in terms of MAP@tIoU on ActivityNet-1.3.
Although SRG achieved the best performances at low and middle tIoU values in Fig. 8, there is still room for improvement at high tIoU values larger than 0.80. Specifically, TIGN in SRG aims to generate as many temporal intervals with reliable boundaries as possible for avoiding missing action instances. Thus, the generated temporal intervals have similar boundaries that are highly overlapped with true action instances. Since the boundary of action is subjective and ambiguous, a large number of the highly overlapped temporal intervals make it difficult for TIEN to precisely rank them. For example, it is a challenging task to find the best temporal intervals when they all have high tIoU values. In a similar vein, we can explain why SRG achieved lower performance than BSN [22] at the tIoU value of 0.95 and outperformed other methods at tIoU values of 0.50 and 0.75, as shown in Table VIII. For future work, we intend to address this limitation of SRG.

In Table II, SRG achieves the best performance in terms of AR@AN=100, but SRG shows comparable performance in terms of the AUC. As aforementioned, TIGN generates a large number of temporal intervals with high tIoU values. However, TIEN struggles to precisely rank these temporal intervals when most of the temporal intervals have good-enough boundaries. This limitation of SRG yields low AR values at small AN values; thus, the AUC of SRG is slightly inferior to those of BSN and MGG [23]. Additionally, inaccurate relatedness score maps for some short action instances on ActivityNet-1.3 (see Fig. 11) would limit the performance increase. Improving these relatedness score maps will be another future work.

VI. CONCLUSION

In this paper, we have proposed a novel approach, namely SRG, for high-quality temporal action proposal generation. SRG is based on two key ideas, “snippet relatedness” and “PN operations”. The snippet relatedness indicates which snippets are related to a specific action instance. PN operations allow the snippet relatedness to be effectively learned by locally and globally capturing long-range dependencies among snippets. By employing these ideas, SRG first produces three types of 2D score maps for temporal intervals. Then, SRG evaluates the action confidence scores of the temporal intervals and refines their boundaries to generate high-quality proposals. Experimentally, we have demonstrated that SRG outperforms state-of-the-art methods on both the THUMOS-14 and ActivityNet-1.3 datasets. Furthermore, with the same standard action classifier, our proposals markedly outperform state-of-the-art proposals for temporal action detection.

REFERENCES

[1] Y. Iwasita, M. Ryoo, T. J. Fuchs, and C. Padgett, “Recognizing humans in motion: Trajectory-based aerial video analysis,” in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2013, pp. 127.1–127.11.
[2] T. Shu, D. Xie, B. Rothrock, S. Todorovic, and S. C. Zhu, “Joint inference of groups, events and human roles in aerial videos,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4756–4854.
[3] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, “Describing videos by exploiting temporal structure,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 199–211.
[4] T. Yao, T. Mei, and Y. Rui, “Highlight detection with pairwise deep ranking for first-person video summarization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 982–990.
[5] K. Zhang, W.-L. Chao, and K. Grauman, “Video summarization with long short-term memory,” in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 766–782.
[6] C.-L. Chou, H.-T. Chen, and S.-Y. Lee, “Pattern-based near-duplicate video retrieval and localization on web-scale videos,” *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 382–395, Jan. 2015.
[7] M. Douze, J. Revaud, J. Varbeek, H. Jégou, and C. Schmid, “Circulant temporal encoding for video retrieval and temporal alignment,” *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 291–306, Sep. 2016.
[8] J. Song, L. Gao, L. Liu, X. Zhu, and N. Sebe, “Quantization-based hashing: A general framework for scalable image and video retrieval,” *Pattern Recognit.*, vol. 75, pp. 175–187, Mar. 2018.
[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2900–2908.
[11] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2015, pp. 91–99.
[12] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, “Turn tap: Temporal unit regression network for temporal action proposals,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3648–3656.
[13] F. C. Heilbron, J. C. Niebles, and B. Ghahem, “Fast temporal activity proposals for efficient detection of human actions in untrimmed videos,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1914–1923.
[14] Z. Shou, D. Wang, and S.-F. Chang, “Temporal action localization in untrimmed videos via multi-stage cnns,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1049–1058.
[15] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, “Rethinking the faster r-cnn architecture for temporal action localization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1130–1139.
[16] F. C. Heilbron, W. Barrios, V. Escorcia, and B. Ghahem, “Ssc: Semantic context cascade for efficient action detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3175–3184.
[17] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng, “Temporal action localization by structured maximal sums,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3684–3692.
[18] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, “Temporal action detection with structured segment networks,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2914–2923.
[19] S. Bach, V. Escorcia, C. Shen, B. Ghahem, and J. C. Niebles, “Sst: Single-stream temporal action proposals,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6373–6382.
[20] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, “Daps: Deep action proposals for action understanding,” in Proc. Eur. Conf. Comput. Vis., Oct. 2016, pp. 768–784.

[21] J. Gao, K. Chen, and R. Nevatia, “Ctap: Complementary temporal action proposal generation,” in Proc. Eur. Conf. Comput. Vis., Sep. 2018, pp. 68–83.

[22] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, “Bsn: Boundary sensitive network for temporal action proposal generation,” in Proc. Eur. Conf. Comput. Vis., Sep. 2018, pp. 3–19.

[23] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S.-F. Chang, “Multi-granularity generator for temporal action proposal,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 3604–3613.

[24] Q. Wu, H. Li, K. N. Ngan, and K. Ma, “Image quality assessment using local consistency aware retriever and uncertainty aware evaluator,” IEEE Trans. Circuits Syst. Video Technol., vol. 28, no. 9, pp. 2078–2089, Sep. 2018.

[25] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, “Learning blind video temporal consistency,” in Proc. Eur. Conf. Comput. Vis., Sep. 2018, pp. 179–195.

[26] R. Dechter and P. van Beek, “Local and global relational consistency,” Theor. Comput. Sci., vol. 173, no. 1, pp. 283–308, Feb. 1997.

[27] Y. G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, “Thumos challenge: Action recognition with a large number of classes,” 2014, http://iccv.ucsf.edu/THUMOS14/.

[28] F. C. Heilbron, B. G. V. Escorcia, and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 961–970.

[29] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, “Recognizing action at a distance,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2003, pp. 726–733.

[30] K. Jia and D.-Y. Yeung, “Human action recognition using local spatio-temporal discriminant embedding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.

[31] H. Wang, A. Kläser, C. Schmid, and C. L. Liu, “Action recognition by dense trajectories,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2011, pp. 3169–3176.

[32] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 4724–4733.

[33] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 1933–1941.

[34] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 4694–4702.

[35] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Proc. Adv. Neural Inf. Process. Syst., Dec. 2014, pp. 568–576.

[36] L. Wang, Y. Xiong, Z. Wang, Y. Q. D. Lin, X. Tang, and L. van Gool, “Temporal segment networks: Towards good practices for deep action recognition,” in Proc. Eur. Conf. Comput. Vis., Oct. 2016, pp. 29–36.

[37] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6546–6555.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 4489–4497.

[39] H. Xu, A. Das, and K. Saenko, “R-c3d: Region convolutional 3d network for temporal activity detection,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 5783–5792.

[40] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, “Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 5734–5743.

[41] J. Gao, Z. Yang, and R. Nevatia, “Cascaded boundary regression for temporal action detection,” in Proc. Brit. Mach. Vis. Conf., Sep. 2017, pp. 52.1–52.11.

[42] J. B. Roerdink and A. Meijster, “The watershed transform: Definitions, algorithms and parallelization strategies,” Fund. Informaticae, vol. 41, pp. 187–228, Apr. 2000.

[43] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7132–7141.

[44] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in Proc. Eur. Conf. Comput. Vis., Sep. 2018, pp. 3–19.

[45] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7794–7803.

[46] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 2881–2890.

[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 2818–2826.

[48] T. Lin, X. Zhao, and Z. Shou, “Single shot temporal action detection,” in Proc. ACM Multimedia, Oct. 2017, pp. 988–996.

[49] K. Yang, P. Qiao, D. Li, S. Lv, and Y. Dou, “Exploring temporal preservation networks for precise temporal action localization,” in Proc. AAAI Conf. Artif. Intell., Feb. 2018, pp. 7477–7484.

[50] H. Qi, Y. Zheng, H. Ye, Y. Lu, F. Wang, and L. He, “Precise temporal action localization by evolving temporal proposals,” in Proc. Int. Conf. Multimdia Retr., Jun. 2018, pp. 388–396.

[51] Q. Wu, H. Li, F. Meng, and K. N. Ngan, “Generic proposal evaluator: A lazy learning strategy toward blind proposal quality assessment,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 1, pp. 306–319, Jan. 2018.

[52] L. Wang, Y. Xiong, D. Lin, and L. van Gool, “Untrimmednets for weakly supervised action recognition and detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 4325–4334.

[53] Y. Xiong, L. Wang, Z. Wang, B. Zhang, H. Song, W. Li, D. Lin, Y. Qiao, L. van Gool, and X. Tang, “Cuhk & ethz & siat submission to activitynet challenge 2016,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, Jun. 2016.

[54] T. Lin, H. Su, and X. Zhao, “Boundary sensitive network: Submission to activitynet challenge 2018,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, Jun. 2018.