BSN++: Complementary Boundary Regressor with Scale-Balanced Relation Modeling for Temporal Action Proposal Generation

Haisheng Su¹, Weihao Gan¹, Wei Wu¹, Junjie Yan¹, Yu Qiao²

¹ SenseTime Group Limited
² Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
{suhaiseng, ganweihao, wuwei, yanjunjie}@sensetime.com, yu.qiao@siat.ac.cn

Abstract

Generating human action proposals in untrimmed videos is an important yet challenging task with wide applications. Current methods often suffer from the noisy boundary locations and the inferior quality of confidence scores used for proposal retrieving. In this paper, we present BSN++, a new framework which exploits complementary boundary regressor and relation modeling for temporal proposal generation. First, we propose a novel boundary regressor based on the complementary characteristics of both starting and ending boundary classifiers. Specifically, we utilize the U-shaped architecture with nested skip connections to capture rich contexts and introduce bi-directional boundary matching mechanism to improve boundary precision. Second, to account for the proposal-proposal relations ignored in previous methods, we devise a proposal relation block to which includes two self-attention modules from the aspects of position and channel. Furthermore, we find that there inevitably exists data imbalanced problems in the positive/negative proposals and temporal durations, which harm the model performance on tail distributions. To relieve this issue, we introduce the scale-balanced re-sampling strategy. Extensive experiments are conducted on two popular benchmarks: ActivityNet-1.3 and THUMOS14, which demonstrate that BSN++ achieves the state-of-the-art performance.

Introduction

Temporal action detection task has received much attention from many researchers in recent years, which requires not only categorizing the real-world untrimmed videos but also locating the temporal boundaries of action instances. Akin to object proposals for object detection in images, temporal action proposal indicates the temporal intervals containing the actions and plays an important role in temporal action detection. It has been commonly recognized that high-quality proposals usually have two crucial properties: (1) the generated proposals should cover the action instances temporally with both high recall and temporal overlapping; (2) the quality of proposals should be evaluated comprehensively and accurately, thus providing a overall confidence for later retrieving step.

To cater for these two conditions and achieve high quality proposals, there are two main categories in the existing proposal generation methods (Buch et al. 2017b; Gao et al. 2017; Lin, Zhao, and Shou 2017a; Shou, Wang, and Chang 2016). The first type adopts the top-down fashion, where proposals are generated based on sliding windows (Shou, Wang, and Chang 2016) or uniform-distributed anchors (Lin, Zhao, and Shou 2017a), then a binary classifier is employed to evaluate confidence for the proposals. However, the proposals generated in this way are doomed to have imprecise boundaries though with regression. Under this circumstance, the other type of methods (Lin et al. 2018; Xiong et al. 2017; Lin et al. 2019) attract many researchers recently which tackle this problem in a bottom-up fashion, where the input video is evaluated in a finer-level.
Sensitive Network (BSN) to generate proposals with flexible durations and reliable confidence scores. Though BSN achieves convincing performance, it still suffers from three main drawbacks: (1) BSN only employs the local details around the boundaries to predict boundaries, without taking advantage of the rich temporal contexts through the whole video sequence; (2) BSN fails to consider the proposal-proposal relations for confidence evaluation; (3) the imbalance data distribution between positive/negative proposals and temporal durations is also neglected.

To relieve these issues, we propose BSN++, for temporal proposal generation. (i) To exploit the rich contexts for boundary prediction, we adopt the U-shaped architecture with nested skip connections. Meanwhile, the two optimized boundary classifiers share the same goals especially in detecting the sudden change from background to actions or learning the discriminativeness from actions to background, thus are complementary with each other. Under this circumstance, we propose the complementary boundary regressor, where the starting classifier can also be used to predict the ending locations when the input videos are processed in a reversed direction, and vice versa. In this way, we can achieve high precision without adding extra parameters. (ii) In order to predict the confidence scores of densely-distributed proposals, we design a proposal relation block aiming at levering both channel-wise and position-wise global dependencies for proposal-proposal relation modeling. (iii) To relieve the imbalance scale-distribution among the sampling positives as well as the negatives (see Fig. 1), we implement a two-stage re-sampling scheme consisting of the IoU-balanced (positive-negative) sampling and the scale-balanced re-sampling. The boundary map and the confidence maps are generated simultaneously and jointly trained in a unified framework. In summary, the main contributions of our work are listed below in three-folds:

- We revisit the boundary prediction problem and propose a complementary boundary generator to exploit both “local and global”, “past and future” contexts for accurate temporal boundary prediction.
- We propose a proposal relation block for proposal confidence evaluation, where two self-attention modules are adopted to model the proposal relations from two complementary aspects. Besides, we devise a two-stage re-sampling scheme for equivalent balancing.
- Thorough experiments are conducted to reveal the effectiveness of our method. Further combining with the existing action classifiers, our method can achieve the state-of-the-art temporal action detection performance.

**Related Work**

**Action Recognition**

Action recognition is an essential branch which has been extensively explored in recent years. Earlier methods such as improved Dense Trajectory (iDT) (Wang et al. 2011) Wang and Schmid 2013) mainly adopt the hand-crafted features including HOG, MBH and HOF. Current deep learning based methods (Feichtenhofer, Pinz, and Zisserman 2016 Simonyan and Zisserman 2014, Tran et al. 2015, Wang et al. 2016) typically contain two main categories: the two-stream networks (Feichtenhofer, Pinz, and Zisserman 2016 Simonyan and Zisserman 2014) capture the appearance features and motion information from RGB image and stacked optical flow respectively; 3D networks (Tran et al. 2015 Qiu, Yao, and Tao 2017) exploit 3D convolutional layers to capture the spatial and temporal information directly from the raw videos. Action recognition networks are usually adopted to extract visual feature sequence from untrimmed videos for the temporal action proposals and detection task.

**Imbalanced Distribution Training**

Imbalanced data distribution naturally exists in many large-scale datasets (Cordts et al. 2018, Cordts et al. 2016). Current literature can be mainly divided into three categories: (1) re-sampling, includes oversampling the minority classes (Estabrooks, Jo, and Japkowicz 2004) or downsampling the majority classes (Weiss, McCarthy, and Zabar 2007); (2) re-weighting, namely cost sensitive learning (McCarthy, Zabar, and Weiss 2005, Cui et al. 2019), which aims to dynamically adjust the weight of samples or different classes during training process. (3) In object detection task, the imbalanced data distribution is more serious between background and foreground for one-stage detector. Some methods such as Focal loss (Lin et al. 2017) and online hard negative mining (Shrivastava, Gupta, and Girshick 2016) are designed for two-stage detector. In this paper, we implement the scale-balanced re-sampling upon the IoU-balanced sampling for proposal confidence evaluation, motivated by the mini-batch imbalanced loss distribution against proposal durations.

**Temporal Action Detection and Proposals**

Akin to object detection in images, temporal action detection also can be divided into proposal and classification stages. Current methods train these two stages separately (Shou, Wang, and Chang 2016, Singh and Cuzzolin 2016) or jointly (Buch et al. 2017a, Lin, Zhao, and Shou 2017a). As for proposal generation, top-down methods (Shou, Wang, and Chang 2016) (Lin, Zhao, and Shou 2017a) are mainly based on sliding windows or pre-defined anchors, while bottom-up methods (Xiong et al. 2017, Lin et al. 2018, 2019) first evaluate the actionness or boundary probabilities of each temporal location in a finer level. However, proposals generated in a local fashion of (Xiong et al. 2017) cannot be further retrieved without confidence scores evaluated from a global view. And probabilities sequence generated in (Lin et al. 2018, 2019) (Liu et al. 2019) is sensitive to noises, causing many false alarms. Besides, proposal-proposal relations fail to be considered for confidence evaluation. Meanwhile, the imbalanced distribution among the proposals remains to be settled. To address these issues, we propose BSN++, which is unique to previous works in three main aspects: (1) we revisit the boundary prediction task and propose a complementary boundary generator to exploit rich contexts together with bi-directional matching strategy for boundary prediction; (2) we propose a proposal relation block for proposal-proposal relations modeling; (3) two-stage re-sampling scheme is designed for equivalent balancing.
Our Approach

Problem Definition

Denote an untrimmed video sequence as \( U = \{ u_i \}_{i=1}^{l_v} \), where \( u_i \) indicates the \( t \)-th frame in the video of length \( l_v \). A set of action instances \( \Psi_g = \{ \varphi_n = (t_n^s, t_n^e) \}_{n=1}^{N_g} \) are temporally annotated in the video \( S_v \), where \( N_g \) is the number of ground truth action instances, and \( t_n^s, t_n^e \) are the starting time and ending time of the action instance \( \varphi_n \) respectively. During training phase, the \( \Psi_g \) is provided. While in the testing phase, the predicted proposal set \( \Psi_p \) should cover the \( \Psi_g \) with high recall and high temporal overlapping.

Video Feature Encoding

Before applying our algorithm, we adopt the two-stream network (Simonyan and Zisserman 2014) in advance to encode the visual features from raw video as many previous works (Lin et al. 2018; Gao, Kan, and Nevatia 2018; Su, Zhao, and Lin 2018). This kind of architecture has been widely used in many video analysis tasks (Lin, Zhao, and Shou 2017; Zhao et al. 2017; Gao, Yang, and Nevatia 2017). Concretely, given an untrimmed video \( S_v \), which contains \( l_v \) frames, we process the input video in a regular interval \( \sigma \) for reducing the computational cost. We concatenate the output of the last FC-layer in the two-stream network to form the feature sequence \( F = \{ f_i \}_{i=1}^{l_v/\sigma} \), where \( l_v = l_v/\sigma \). Final, the feature sequence \( F \) is used as the input of our BSN++.

Proposed Network Architecture: BSN++

In contrast to the previous BSN (Lin et al. 2018), which consists of multiple stages, BSN++ is designed to generate the proposal map directly in a unified network. To obtain the proposal map, BSN++ first generates the boundary map which represents the boundary information and confidence map which represents the confidence scores of densely distributed proposals. As shown in Fig. 2, BSN++ model mainly contains three main modules: Base Module handles the input video features to perform temporal information modeling, then the output features are shared by the two following modules. Complementary Boundary Generator processes the input video features to evaluate the boundary probabilities sequence, using a nested U-shaped encoder-decoder; Proposal Relation Block aims to model the proposal-proposal relations with two self-attention modules responsible for two complementary dependencies.

Base Module. The goal of this module is to handle the extracted features for temporal relationship modeling, which serves as the base module of the following two branches. It mainly includes two 1D convolutional layers, with 256 filters, kernel size 3 and stride 1, followed by a ReLU activation layer. Since the length of videos is uncertain, we truncate the video sequence into a series of sliding windows. The detailed data construction is illustrated in Section 4.1.

Complementary Boundary Generator. Inspired by the success of U-Net (Ronneberger, Fischer, and Brox 2015; Zhou et al. 2018) used in image segmentation, we design our boundary generator as Encoder-Decoder networks because this kind of architecture is able to capture both high-level global context and low-level local details at the same time. As shown in Fig. 3, each circle represents a 1D convolutional layer with 512 filters and kernel size 3, stride 1, together with a batch normalization layer and a ReLU layer except the prediction layer. To reduce over-fitting, we just add two down-sampling layers to expand the receptive fields.
and the same number of up-sampling layers are followed to recover the original temporal resolutions. Besides, deep supervision (shown red) is also performed for fast convergent speed and nested skip connections are employed for bridging the semantic gap between feature maps of the encoder and decoder prior to fusion. We observe that the starting classifier learns to detect the sudden change from background to actions and vice versa. Hence, the starting classifier can be regarded as a pseudo ending classifier when processes the input video in a reversed direction, thus the bi-directional prediction results are complementary. With this observation, bi-directional encoder-decoder networks are optimized in parallel, and the consistent constraint is performed upon the intermediate features (i.e. $f(x^i_T)$ and $f(x^i_h)$) on both sides before the prediction layer as shown in Fig. 3. During the inference stage, the aforementioned encoder-decoder network is adopted to predict the the starting heatmap $\hat{H}^s = \{\hat{h}^s_i\}_{i=1}^N$ and ending heatmap $\hat{H}^e = \{\hat{h}^e_i\}_{i=1}^N$, respectively, where $h^s_i$ and $h^e_i$ indicate the starting and ending probabilities of the i-th snippet respectively. Meanwhile, we feed the input feature sequence in a reversed order to the identical backbone. Similarly, we can obtain the starting heatmap $\hat{H}^s$ and ending heatmap $\hat{H}^e$.

After the two-passes, in order to select the boundaries of high scores, we fuse the two pairs of heatmaps to yield the final heatmaps:

$$H^s = \{\sqrt{h^s_i \times h^l_i}\}_{i=1}^N, \quad H^e = \{\sqrt{h^e_i \times h^l_i}\}_{i=1}^N.$$  \hspace{1cm} (1)

With these two boundary points heatmaps, we can further construct the boundary map $M^b = \mathbb{R}^{1 \times D \times T}$ which can represent the boundary information of all densely distributed proposals, where $T$ and $D$ are the length of the feature sequence and maximum duration of proposals separately:

$$M^b_{i,j} = \{(h^s_i \times h^l_{i+j})_{j=1}^D\}_{i=1}^N, \quad i + j < T, \quad (2)$$

**Proposal Relation Block.** The goal of this block is to evaluate the confidence scores of dense proposals. Before performing proposal-proposal relations, we follow the previous work BMN [Lin et al. 2019] to generate the proposal feature maps as $F^P = \mathbb{R}^{D \times T \times N}$. N is set to 32. Then the proposal feature maps are fed to a 3D convolutional layer with kernel size $1 \times 1 \times 32$ and 512 filters, followed by a ReLU activation layer. Thus the reduced proposal features maps are $F^p = \mathbb{R}^{D \times T \times 512}$. The proposal relation block consists of two self-attention modules as follows.

**Position-aware attention module.** As illustrated in Fig. 4, given the proposal feature maps $F^p$, we adopt the similar self-attention architecture as Non-local network [Wang et al. 2018], where the proposal feature maps are fed into a convolutional layer separately to generate two new feature maps $A$ and $B$ for spatial matrix multiplication with reshape and transpose operations. And then a Softmax layer is applied to calculate the position-aware attention $P^A = \mathbb{R}^{L \times L}$, where $L = D \times T$:

$$P^A_{j,i} = \frac{\exp(A_i \cdot B_j)}{\sum_{j=1}^{L} \exp(A_i \cdot B_j)}, \hspace{1cm} (3)$$

Figure 4: Illustration of the proposal relation block. After generating the proposal feature maps, two complementary branches are followed to model the proposal relation separately. In the upper branch, the position-aware attention module aims to leverage global dependencies. While in the lower branch, the channel-aware attention module aims to attend to the discriminative features by channel matrix calculation. Finally, we aggregate the outputs from the three branches for pixel-level confidence prediction.

**Channel-aware attention module.** In contrast to the position-aware attention module, this module directly performs channel-wise matrix multiplication in order to exploit the inter-dependencies among different channels, which can help enhance the proposal feature representations for confidence prediction. The process of attention calculation is the same as the former module except for the attended dimension. Similarly, the attended features after weighted summed with the proposal features are further captured by a 2D convolutional layer to generate the confidence map $M^c = \mathbb{R}^{D \times T}$. We also aggregate the outputs of the two attention modules for proposal confidence prediction, and finally we fuse the predicted confidence maps from the three branches for a better performance.

**Re-sampling.** Imbalanced data distribution can affect the model training especially in the long-tailed dataset. In this paper, we revisit the positive/negative samples distribution for improving the quality of proposal confidence prediction and design a proposal-level re-sampling method to improve the performance of training on the long-tailed dataset. Our re-sampling scheme consists of two stages aiming at not only balancing the positives and negatives, but also balancing the temporal duration of the proposals.

**IoU-balanced sampling** As shown in Fig. 1, we can see from the mini-batch loss distribution that the number of positives and negatives differs greatly which dooms to bias the training model without effective measures. Previous works usually design a positive-negative sampler (i.e. IoU-balanced sampler) to balance the data distribution for each mini-batch, thus ensuring the ratio of positive and negative
samples is nearly 1:1. However, we can also conclude from the Fig. 1 that the scale of positives or negatives fails to conform the uniform distribution. Under this circumstance, we should consider how to balance the scales of proposals.

### Scale-balanced re-sampling

To relieve the issue among long-tailed scales, we propose a second-stage positive/negative re-sampling method, which is upon the principle of IoU-balanced sampling. Specifically, define \( P_i \) as the number of positive proposals with the scale \( s_i \), then \( r_i \) is the positive ratio of \( s_i \):

\[
    r_i = \frac{P_i}{\sum_{j=1}^{N_s} P_j}.
\]

\[
    r_i' = \begin{cases} \lambda \cdot \exp\left( \frac{r_i}{\lambda} \right) & (0 < r_i \leq \lambda), \\ r_i & (\lambda < r_i \leq 1), \end{cases}
\]

where \( N_s \) is the number of pre-defined normalized scale regions (i.e. \([0 - 0.3, 0.3 - 0.7, 0.7 - 1.0]\)). Then we design a positive ratio sampling function, the resulting ratio \( r_i' \) is bigger than \( r_i \) for proposal scale with a frequency lower than \( \lambda \), where \( \lambda \) is a hyper-parameter which we set to 0.15 empirically. Hence, we use the re-normalized \( r_i \) as the sampling probability of the specific proposal scale region \( s_i \) to construct the mini-batch data. As for the negative proposals, the same process is performed.

### Training and Inference of BSN++

In this section, we will introduce the training and inference details of BSN++.

#### Training

### Overall Objective Function.

As described above, BSN++ consists of three main sub-modules. The multi-task objective function is defined as:

\[
    L_{BSN++} = L_{CBG} + \beta \cdot L_{PRB} + \gamma \cdot L_2(\Theta),
\]

where \( L_{CBG} \) and \( L_{PRB} \) are the objective functions of the complementary boundary generator and the proposal relation block respectively, while \( L_2(\Theta) \) is a regularization term. \( \beta \) and \( \gamma \) are set to 10 and 0.0001 separately to trade off the training process of two modules and reduce over-fitting.

### Training Data Construction.

Given the extracted feature sequence \( F \) with length \( L_w \), we truncate \( F \) into sliding windows of length \( l_w \) with 75% temporal over-lapping. Then we construct the training dataset as \( \Phi = \{ (F_{w}^n)_{n=1}^{N_w} \} \), where \( N_w \) is the number of retained windows containing at least one ground-truth.

### Label Assignment.

For the Complementary Boundary Generator (CBG), in order to predict the boundary probabilities sequence, we need to generate the corresponding label sequence \( G_w^s \) and \( G_w^e \) as in (Lin et al. 2018). Specifically, for each action instance \( \varphi_g \) in the annotation set \( \Psi_w \), we denote it’s starting and ending regions as \([t_{s}^{g} - d_{c}/10, t_{e}^{g} + d_{c}/10]\) and \([t_{s}^{g} - d_{s}/10, t_{e}^{g} + d_{s}/10]\) respectively, where \( d_{c} = t_{e}^{g} - t_{s}^{g} \) is the duration of \( \varphi_{g} \). Then for each temporal location, if it lies in the starting or ending regions of any action instances, the corresponding label \( g^{s} \) or \( g^{e} \) will be set to 1. Hence the label sequence of starting and ending used in CBG are \( G_{w}^s = \{ g_{w}^s \}_{n=1}^{N_{w}}, G_{w}^e = \{ g_{w}^e \}_{n=1}^{N_{w}} \) respectively.

For the Proposal Relation Block (PRB), we predict the confidence map \( M^c \in R^{D \times I_w} \) of all densely distributed proposals, where the point \( g_{j,i} \) in the label confidence map \( M^c = \{ g_{j,i} \}_{j=1}^{D} \) represents the maximum IoU (Intersection-over-Union) values of proposal \( \varphi_{j,i} = \{ t_{s} = i, t_{e} = i + j \} \) with all \( \varphi_{g} \) in \( \Psi_{g} \).

### Objective of CBG.

We follow (Lin et al. 2018) to adopt the weighted binary logistic regression loss \( L_{cbg} \) as the objective between the output probability and the corresponding label sequence. The objective is:

\[
    L_{CBG} = L_{fwd}^{b} + L_{bkwd}^{b} + L_{fwd}^{c} + L_{bkwd}^{c} + ||f(x^{f}) - f(x^{b})||^2, \tag{6}
\]

where \( L_{fwd}^{b} \) and \( L_{bkwd}^{b} \) represent the \( L_{bl} \) between \( \tilde{H}^{s} \) and \( G_{s}^{w} \), \( H^{e} \) and \( G_{e}^{w} \) respectively in the forward pass. Mean-Squared Loss is also performed on two-passes intermediate features.

### Objective of PRB.

Taking the constructed proposal feature maps \( F^p \) as input, our PRB will generate two types of confidence maps \( M_{bc}^{c} \) and \( M_{cr}^{c} \) for all densely distributed proposals as (Lin et al. 2019). The training objective is defined as the regression loss \( L_{reg} \) and the binary classification loss \( L_{cls} \) respectively:

\[
    L_{PRB} = L_{reg} + L_{cls}, \tag{7}
\]

where the smooth-L1 loss (Girshick 2015) is adopted as \( L_{reg} \), and the points \( g_{j,i} \) with value larger than 0.7 or lower than 0.3 are regarded as positives and negatives respectively.

And we ensure the scale and number ratio between positives and negatives to be near 1:1 by the two-stage sampling scheme described above.

### Inference

During inference stage, our BSN++ can generate the boundary map \( M^b \) based on the bidirectional boundary probabilities (\( H^{s} \) and \( H^{e} \)) and confidence map (\( M_{bc}^{c} \) and \( M_{cr}^{c} \)). We form the proposal map \( M^p \) directly by fusing the \( M^b \) and \( M^r \) with dot multiplication. Then we can filter the points with high scores in the proposal map \( M^p \) as candidate proposals used for post-processing.

### Score Fusion.

As described above, the final scores of proposals in \( M^p \) involve the local boundary information and global confidence scores. Take the proposal \( \varphi = \{ t_{s}, t_{e} \} \) for example, the combination of final score \( s_{p} \) can be shown as:

\[
    p_{c} = M_{t_{e}^{c} - t_{s}^{c}} \cdot \sqrt{M_{t_{e}^{c} - t_{s}^{c}} \cdot M_{t_{e}^{r} - t_{s}^{r}} \cdot M_{t_{e}^{r} - t_{s}^{r}}}, \tag{8}
\]

### Redundant Proposals Suppression.

With the above process, our BSN++ can generate the proposal candidates set as \( \Psi_{p} = \{ \varphi_{n} = \{ t_{s}, t_{e}, p_{c} \} \}_{n=1}^{N_{p}} \), where \( N_{p} \) is the number of proposals. Since the generated proposals may overlap with each other to various degrees, we conduct SoftNMS (Bodla et al. 2017) algorithm to suppress the confidence scores of redundant proposals. Finally, the proposals set is \( \Psi_{p} = \{ \varphi_{n} = \{ t_{s}, t_{e}, p_{c} \} \}_{n=1}^{N_{p}} \), where \( p_{c} \) is the decayed score of proposal \( \varphi_{n} \).
Experiments

Datasets and Setup

Datasets. ActivityNet-1.3 (Caba Heilbron et al. 2015) is a large-scale video dataset for action recognition and temporal action detection tasks used in the ActivityNet Challenge from 2016 to 2019. It contains 19,994 videos with 200 action classes temporally annotated, and the ratio of training, validation and testing sets is 1:1:2. THUMOS-14 (Jiang et al. 2014) contains 200 and 213 untrimmed videos with temporal annotations of 20 action classes in validation and testing sets respectively, which is commonly used for temporal action detection task.

Implementation details. For feature encoding, we adopt the two-stream network (Simonyan and Zisserman 2014), where ResNet network (He et al. 2016) and BN-Inception network (Ioffe and Szegedy 2015) are used as the spatial and temporal networks respectively. During feature extraction, the interval \( \sigma \) is set to 16 and 5 on ActivityNet-1.3 and THUMOS-14 respectively.

On ActivityNet-1.3, we rescale the feature sequence of input videos to \( l_w = 100 \) by linear interpolation following the previous (Lin et al. 2018), and the maximum duration \( D \) is also set to 100 to cover all action instances. While on THUMOS14, the length \( l_w \) of sliding windows is set to 128 while the maximum duration \( D \) is set to 64, which can cover almost 98% action instances. On both datasets, we train our BSN++ from scratch using the Adam optimizer and the batch size is set to 16. And the initial learning rate is set to 0.001 for 7 epochs, then 0.0001 for another 3 epochs.

Temporal Proposal Generation

Evaluation metrics. Following the conventions, Average Recall (AR) is calculated under different \( \text{IoU} \) thresholds which are set to \([0.5:0.05:0.95]\) on ActivityNet-1.3, and \([0.5:0.05:1.0]\) on THUMOS14. We measure the relation between AR and Average Number (AN) of proposals, denoted as AR@AN. And we also calculate the area (AUC) under the AR vs. AN curve as another evaluation metric on ActivityNet-1.3 dataset, where AN ranges from 0 to 100.

Comparison to the state-of-the-arts. Table 1 illustrates the comparison results on ActivityNet-1.3. It can be observed that our BSN++ outperforms other state-of-the-art proposal generation methods with a big margin in terms of AR@AN and AUC on validation set of ActivityNet-1.3. For a direct comparison to BSN, our BSN++ improves AUC from 66.17% to 68.26% on validation set. Particularly, when the AN is 100, our method significantly improves AR from 74.16% to 76.52% by 2.36%. And when the AN is 1, the AR which our BSN++ can obtain is 34.30%.

Table 2 demonstrates that: (1) the Encoder-Decoder architecture can effectively learn “local and global” contexts for accurate and discriminative proposals scoring; (4) the bidirectional matching mechanism further validates the importance of future context in assisting the boundary judgement; (3) as for the proposal quality evaluation, unlike the previous works which treat the proposals separately, the proposal relation block can provide more global and representative features for accurate and discriminative proposals scoring; (4) besides, with scale-balanced sampling, the model can obtain equivalent balancing; (5) final, integrating all the separated modules into an end-to-end network, we can obtain the competing performance improvement; (6) BSN++ achieves the great overall efficiency than previous methods.

Ablation Experiments

In this section, we comprehensively evaluate our proposed BSN++ on the validation set of ActivityNet-1.3.

Effectiveness and efficiency of modules in BSN++. We perform the ablation studies with different architecture settings to verify the effectiveness and efficiency of each module proposed in BSN++. The evaluation results shown in Table 3 demonstrate that: (1) the Encoder-Decoder architecture can effectively learn “local and global” contexts for accurate and discriminative proposals scoring; (4) besides, with scale-balanced sampling, the model can obtain equivalent balancing; (5) final, integrating all the separated modules into an end-to-end network, we can obtain the competing performance improvement; (6) BSN++ achieves the great overall efficiency than previous methods.

Ablation comparison with BSN. We conduct a direct comparison to BSN (Lin et al. 2018) to confirm the effectiveness and superiority of our BSN++. As shown in Table 3, the TEM of BSN which only considers the local details for boundary probabilities sequence generation is inferior with limited receptive fields. Meanwhile, without the full use of temporal context, it is also not robust in complicated sce-
**Table 3: Ablation experiments in the validation set of ActivityNet-1.3.** Complementary boundary generator is abbreviated as CBG and BBM denotes bi-directional matching. PRB is the proposal relation block and SBS is the scale-balanced sampling. PAM and CAM indicate the two self-attention modules. Inference speed here is the seconds (s) cost $T_{cost}$ for processing a 3-minute videos using a Nvidia 1080-Ti card. e2e denotes the joint training manner.

| Model          | Module          | e2e | AUC  | $T_{cost}$ |
|---------------|----------------|-----|------|------------|
| BSN           | TEM            | -   | 64.80| 0.036      |
| BSN           | TEM+PAM        | ✓   | 74.16| 0.629      |
| BMN           | TEM            | -   | 65.17| 0.035      |
| BMN           | TEM+PAM        | ✓   | 67.10| 0.052      |
| BSN++         | CBG(w/o BBM)   | -   | 66.02| 0.019      |
| BSN++         | CBG(w/ BBM)    | -   | 66.43| 0.025      |
| BSN++         | CBG+PRB (w/o SBS) | ×  | 67.34| 0.054      |
| BSN++         | CBG+PRB (w/o SBS) | ✓  | 67.77| 0.039      |
| BSN++         | CBG+PRB (w/o PAM) | ✓  | 67.99| 0.034      |
| BSN++         | CBG+PRB (w/o CAM) | ✓  | 68.01| 0.035      |
| BSN++         | CBG+PRB        | ✓   | 68.26| 0.039      |

**Table 4: Generalizability evaluation on ActivityNet-1.3.**

| Model          | Seen(validation) | Unseen(validation) |
|---------------|-----------------|--------------------|
| Train Data    | AR@100 | AUC  | AR@100 | AUC  |
| Seen+Unseen   | 72.96/74.56     | 65.02/66.34        | 72.68/74.32        | 65.06/66.37   |
| Seen          | 72.47/74.03     | 64.37/65.87        | 72.46/73.82        | 64.47/65.89   |

**Table 5: Detection results compared with (Shou et al. 2017; Zhao et al. 2017; Lin, Zhao, and Shou 2017a; Lin et al. 2018, 2019) on validation set of ActivityNet-1.3, where our proposals are combined with video-level classification results generated by (Xiong et al. 2016).**

| Method          | Classifier | AR@0.5 | AR@0.75 | AR@0.95 | Average |
|-----------------|------------|--------|---------|---------|---------|
| ActivityNet-1.3 |            |        |         |         |         |
| SSN             | Unet       | 63.16  | 24.97   | 63.16   | 24.97   |
| SSAD            |            | 44.39  | 29.65   | 70.97   | 29.17   |
| BSN             |            | 46.45  | 29.96   | 80.07   | 30.03   |
| BMN             |            | 50.07  | 34.78   | 8.29    | 33.85   |
| Ours            |            | 51.27  | 37.50   | 8.33    | 34.88   |

**Table 6: Detection results compared with (Gao et al. 2017; Lin et al. 2018; Liu et al. 2019; Lin et al. 2019) on testing set of THUMOS14, where video-level classifier (Wang et al. 2017) is combined with proposals generated by BSN++.**

| Method          | Classifier | AR@0.5 | AR@0.75 | AR@0.95 | Average |
|-----------------|------------|--------|---------|---------|---------|
| THUMOS14 (testing) |            |        |         |         |         |
| TURN            | Unet       | 6.3    | 14.1    | 24.5    | 35.3    | 46.3    |
| BSN             |            | 20.0   | 28.4    | 36.9    | 45.0    | 53.5    |
| MGG             |            | 21.3   | 29.5    | 37.4    | 46.8    | 53.9    |
| BMN             |            | 20.5   | 29.7    | 38.8    | 47.4    | 56.0    |
| Ours            |            | 22.8   | 31.9    | 41.3    | 49.5    | 59.9    |

**Conclusion**

We propose BSN++ for temporal action proposal generation. The complementary boundary generator takes the advantage of U-shaped architecture and bi-directional boundary matching mechanism to learn rich contexts for boundary prediction. To model the proposal-proposal relations for confidence evaluation, we devise the proposal relation block which employs two self-attention modules to perform global and inter-dependencies modeling. Meanwhile, we are the first to consider the imbalanced data distribution of proposal durations. Both the boundary map and confidence map can be generated simultaneously in a unified network. Extensive experiments conducted on ActivityNet-1.3 and THUMOS14 datasets demonstrate the effectiveness of our method in both temporal action proposal and detection performance.
References

Bodla, N.; Singh, B.; Chellappa, R.; and Davis, L. S. 2017. Improving Object Detection With One Line of Code. In arXiv preprint arXiv:1704.04503.

Buch, S.; Escorcia, V.; Ghanem, B.; Fei-Fei, L.; and Niebles, J. C. 2017a. End-to-End, Single-Stream Temporal Action Detection in Untrimmed Videos. In Proceedings of the British Machine Vision Conference.

Buch, S.; Escorcia, V.; Shen, C.; Ghanem, B.; and Niebles, J. C. 2017b. Sst: Single-stream temporal action proposals. In CVPR, 6373–6382. IEEE.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In CVPR, 961–970.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2018. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv preprint arXiv:1811.00982, 2018.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. CVPR.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. CVPR.

Estabrooks, A.; Jo, T.; and Japkowicz, N. 2004. A multiple resampling method for learning from imbalanced data sets. Computational intelligence.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In CVPR, 1933–1941.

Gao, J.; Kan, C.; and Nevatia, R. 2018. CTAP: Complementary Temporal Action Proposal Generation. arXiv preprint arXiv:1807.04821.

Gao, J.; Yang, Z.; and Nevatia, R. 2017. Cascaded boundary regression for temporal action detection. arXiv preprint arXiv:1705.01180.

Gao, J.; Yang, Z.; Sun, C.; Chen, K.; and Nevatia, R. 2017. TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals. In ICCV, 3648–3656. IEEE.

Girshick, R. 2015. Fast r-cnn. In ICCV, 1440–1448.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In CVPR, 770–778.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Jiang, Y.; Liu, J.; Zamir, A. R.; Toderici, G.; Laptev, I.; Shah, M.; and Sukthankar, R. 2014. THUMOS challenge: Action recognition with a large number of classes. In Computer Vision-ECCV workshop 2014.

Lin, T.; Liu, X.; Li, X.; Ding, E.; and Wen, S. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. CoRR abs/1907.09702.

Lin, T.; Zhao, X.; and Shou, Z. 2017a. Single shot temporal action detection. In Proceedings of the 2017 ACM on Multimedia Conference, 988–996. ACM.

Lin, T.; Zhao, X.; and Shou, Z. 2017b. Temporal Convolution Based Action Proposal: Submission to ActivityNet 2017. arXiv preprint arXiv:1707.06750.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. arXiv preprint arXiv:1806.02964.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollar, P. 2017. Focal loss for dense object detection. ICCV.

Liu, Y.; Ma, L.; Zhang, Y.; Liu, W.; and Chang, S.-F. 2019. Multi-granularity Generator for Temporal Action Proposal. In CVPR, 3604–3613.

McCarthy, K.; Zabar, B.; and Weiss, G. 2005. Does cost-sensitive learning beat sampling for classifying rare classes? Proceedings ofthe 1st international workshop on Utility-based data mining.

Qiu, Z.; Yao, T.; and Tao, M. 2017. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In ICCV, 5534–5542.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, 234–241. Springer.

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S.-F. 2017. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In CVPR, 1417–1426. IEEE.

Shou, Z.; Wang, D.; and Chang, S.-F. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In CVPR, 1049–1058.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. CVPR.

Simonyan, K.; and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In Advances in neural information processing systems, 568–576.

Singh, G.; and Cuzzolin, F. 2016. Untrimmed video classification for activity detection: submission to activitynet challenge. arXiv preprint arXiv:1607.01979.

Su, H.; Zhao, X.; and Lin, T. 2018. Cascaded Pyramid Mining Network for Weakly Supervised Temporal Action Localization. In ACCV.

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

Wang, H.; Kläser, A.; Schmid, C.; and Liu, C.-L. 2011. Action recognition by dense trajectories. In CVPR, 3169–3176. IEEE.

Wang, H.; and Schmid, C. 2013. Action recognition with improved trajectories. In ICCV, 3551–3558.
Wang, L.; Xiong, Y.; Lin, D.; and Van Gool, L. 2017. Untrimmednets for weakly supervised action recognition and detection. In CVPR, volume 2.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In ECCV, 20–36. Springer.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local Neural Networks. CVPR.

Weiss, G. M.; McCarthy, K.; and Zabar, B. 2007. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? Dmin.

Xiong, Y.; Wang, L.; Wang, Z.; Zhang, B.; Song, H.; Li, W.; Lin, D.; Qiao, Y.; Gool, L. V.; and Tang, X. 2016. CUHK & ETHZ & SIAT submission to ActivityNet challenge 2016. CVPR ActivityNet Workshop.

Xiong, Y.; Zhao, Y.; Wang, L.; Lin, D.; and Tang, X. 2017. A pursuit of temporal accuracy in general activity detection. arXiv preprint arXiv:1703.02716.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In ICCV, volume 2.

Zhou, Z.; Siddiquee, M. M. R.; Tajbakhsh, N.; and Liang, J. 2018. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. MICCAI.