# Boundary Proposal Network for Two-Stage Natural Language Video Localization

**Shaoning Xiao,[1] Long Chen,[2*] Songyang Zhang,[3] Wei Ji,[4] Jian Shao,[1] Lu Ye,[5] Jun Xiao[1]**

[1] DCD Lab, College of Computer Science, Zhejiang University
[2] Tencent AI Lab, Shenzhen
[3] University of Rochester
[4] National University of Singapore
[5] Zhejiang University of Science and Technology
{shaoningx, jiwei, jshao, junx}@zju.edu.cn, zjuchenlong@gmail.com, szhang83@ur.rochester.edu, yelue@zust.edu.cn

## Abstract

We aim to address the problem of Natural Language Video Localization (NLVL) — localizing the video segment corresponding to a natural language description in a long and untrimmed video. State-of-the-art NLVL methods are almost in one-stage fashion, which can be typically grouped into two categories: 1) anchor-based approach: it first pre-defines a series of video segment candidates (*e.g.*, by sliding window), and then does classification for each candidate; 2) anchor-free approach: it directly predicts the probabilities for each video frame[1] as a boundary or intermediate frame inside the positive segment. However, both kinds of one-stage approaches have inherent drawbacks: the anchor-based approach is susceptible to the heuristic rules, further limiting the capability of handling videos with variant length. While the anchor-free approach fails to exploit the segment-level interaction thus achieving inferior results. In this paper, we propose a novel *Boundary Proposal Network (BPNet)*, a universal two-stage framework that gets rid of the issues mentioned above. Specifically, in the first stage, BPNet utilizes an anchor-free model to generate a group of high-quality candidate video segments with their boundaries. In the second stage, a visual-language fusion layer is proposed to jointly model the multi-modal interaction between the candidate and the language query, followed by a matching score rating layer that outputs the alignment score for each candidate. We evaluate our BPNet on three challenging NLVL benchmarks (*i.e.*, Charades-STA, TACoS and ActivityNet-Captions). Extensive experiments and ablative studies on these datasets demonstrate that the BPNet outperforms the state-of-the-art methods.

## Introduction

Understanding video content with the aid of natural language, *e.g.*, describing the video content by natural language or grounding language in the video, has drawn considerable interest in both computer vision and natural language processing communities. This kind of task is challenging since it needs to not only understand the video and the sentence separately but also their corresponding interaction. Recently, a core task of this area called **Natural Language Video Localization (NLVL)** (Gao et al. 2017; Hendricks et al. 2017) has been proposed. As shown in Figure 1 (a), given an untrimmed video and a natural language query, NLVL aims to localize the video segment relevant to the query by determining the start point and the end point. NLVL is a worth exploring task due to its potential applications, *e.g.*, video content retrieval (Shao et al. 2018) and video question answering (Lei et al. 2018; Xiao et al. 2020; Ye et al. 2017).

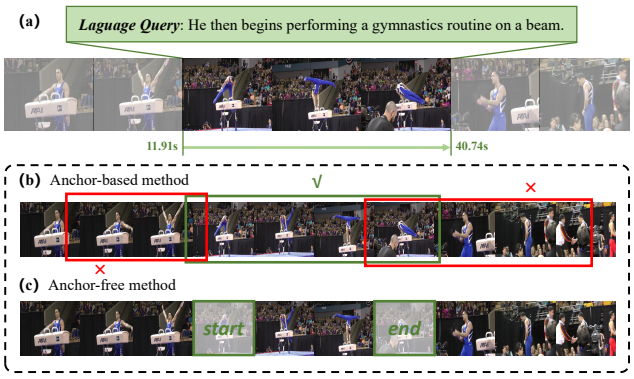


Figure 1: (a) An illustrative example of the NLVL task. Given a video and a query, NLVL is to localize the video segment corresponding to the query with the start point (11.91s) and the end point (40.74s). (b) Anchor-based approach: A number of temporal bounding boxes are placed on the video as candidates and the best-matching one (*e.g.*, the green one) is chosen as the result. (c) Anchor-free approach: Each frame is determined whether it is the boundary frame.

A straightforward solution for NLVL is the *anchor-based approach* (Gao et al. 2017; Hendricks et al. 2018; Liu et al. 2018b; Chen and Jiang 2019; Ge et al. 2019; Xu et al. 2019; Zhang et al. 2019; Chen et al. 2018; Wang, Ma, and Jiang 2020), which follows the same spirit of anchor-based object detectors, *e.g.*, Faster R-CNN (Ren et al. 2015). Specifically, this kind of method places a number of size-fixed bounding boxes on the temporal dimension of the video, and then matches each candidate with the sentence in a common feature space, as shown in Figure 1 (b). They model the segment-level information of proposals and transforms the localization problem into a multi-modal matching problem.

*Long Chen is the corresponding author.



[1] In this paper, the frame is a general concept for an actual video frame or a video clip which consists of a few consecutive frames.

However, it is worth noting that they suffer from two inherent drawbacks: (1) In order to achieve higher recall, a vast number of proposals are required, which makes the subsequent matching process inefficient. (2) Moreover, they have to elaborately design a series of hyper-parameters (*e.g.*, the temporal scales and sample rate) of the bounding boxes so that they can be adaptable to video segments with arbitrary length.

To tackle these issues, another type of solution called the *anchor-free approach* (Chen et al. 2020; Chen and Jiang 2019; Yuan, Mei, and Zhu 2019; Lu et al. 2019; Zhang et al. 2020a; Opazo et al. 2020; Mun, Cho, and Han 2020) has been proposed. As shown in Figure 1(c), instead of depicting the probable video segments as temporal bounding boxes, anchor-free methods directly predict the start and the end boundaries of the query-related video segment or predict the positive frames between the ground-truth boundaries. Benefit from this design, the anchor-free methods get rid of placing superfluous temporal anchors, *i.e.*, they are more computation-efficient. They are also flexible to adapt to diverse video segments without the assumption of the position and the length of the ground-truth video segment. Unfortunately, despite these advantages, there is one main factor that strictly limits the performance of anchor-free methods: they overlook the rich information between start and end boundaries because they are hard to model the segment-level interaction.

In this paper, we propose a two-stage end-to-end framework termed *Boundary Proposal Network (BPNet)*, which inherits the merits of both the anchor-based and anchor-free methods and avoids their defects. Specifically, we first generate several high-quality segment proposals using an anchor-free backbone to avoid redundant candidates, then an individual classifier is proposed to match the proposals with the sentence by predicting the matching score. Compared to anchor-based methods with abundant handcrafted proposals, our design decreases the number of candidates thus alleviating the redundant computation burden. By utilizing an anchor-free method to generate proposals, our approach can be adaptable to video segments of arbitrary lengths without designing the heuristic rules. In addition, compared with anchor-free methods, our BPNet is capable of better modeling the segment-level information via a visual-language fusion module. Furthermore, our proposed framework is a universal paradigm, which means each stage of the framework can be replaced by any stronger anchor-free and anchor-based models to further boost the performance.

We demonstrate the effectiveness of BPNet on three challenging NLVL benchmarks (*i.e.*, TACoS, Charades-STA, and ActivityNet Captions) by extensive ablative studies. Particularly, BPNet achieves new state-of-the-art performance over all three datasets and evaluation metrics.

## Related Work

**Natural Language Video Localization.** The task of natural language video localization (NLVL) aims at predicting the start and end time of the video moment depicted by a language query within the untrimmed video, which was introduced in (Hendricks et al. 2017; Gao et al. 2017). Current existing methods can be roughly grouped into two categories according to how the video segments are detected, namely *anchor-based* methods and *anchor-free* methods.

The anchor-based approaches (Gao et al. 2017; Hendricks et al. 2017, 2018; Liu et al. 2018b,a; Xu et al. 2019; Zhang et al. 2019) solve the NLVL task by matching the predefined video moment proposals (*e.g.*, in sliding window manner) with the language query and choose the best matching video segment as the final result. Gao et al. (2017) proposed a Cross-modal Temporal Regression Localizer (CTRL) model. It takes video moments predefined through sliding windows as input and jointly models text query and video clips, then outputs alignment scores and action boundary regression results for candidate clips. Hendricks et al. (2017) proposed the Moment Context Network (MCN) which effectively localizes natural language queries in videos by integrating local and global video features over time. To improve the performance of the anchor-based method, some works devote to improve the quality of the proposals. Xu et al. (2019) injected text features early on when generating clip proposals to eliminate unlikely clips and thus speed up processing and boost performance. Zhang et al. (2019) proposed to explicitly model moment-wise temporal relations as a structured graph and devised an iterative graph adjustment network to jointly learn the best structure in an end-to-end manner. The others mainly worked on designing a more effective multi-modal interaction network. Liu et al. (2018b) utilized a language-temporal attention network to learn the word attention based on the temporal context information in the video. Liu et al. (2018a) designed a memory attention model to dynamically compute the visual attention over the query and its context information. However, these models are sensitive to the heuristic rules (*e.g.*, the number and the size of anchors) and suffer from inefficiency because of the dense sampling video segment candidates.

The anchor-free approaches (Yuan, Mei, and Zhu 2019; Lu et al. 2019; Chen et al. 2020, 2018; Zhang et al. 2020a) directly predict the probabilities for each frame whether the frame is the boundary frame of the ground-truth video segment. Without pre-defined size-fixed candidates, anchor-free approaches are flexible to adapt to the videos with variant length. Yuan, Mei, and Zhu (2019) directly regressed the temporal coordinates from the global attention outputs. Zhang et al. (2020a) regarded the NLVL task as a span-based QA problem by treating the input video as a text passage and directly regressed the start and end points. In order to further improve the performance, some works focus on eliminating the problem of imbalance of the positive and negative samples. Lu et al. (2019) and Chen et al. (2020) regarded all frames falling in the ground truth segment as foreground, and each foreground frame regresses the unique distances from its location to bi-directional ground truth boundaries. Our BPNet focuses on the other weakness of the anchor-free approach that it is hard to model the segment-level multi-modal features. BPNet takes both frame-level and segment-level visual information into consideration to further improve the performance.

There are also some other works (He et al. 2019; Wang,

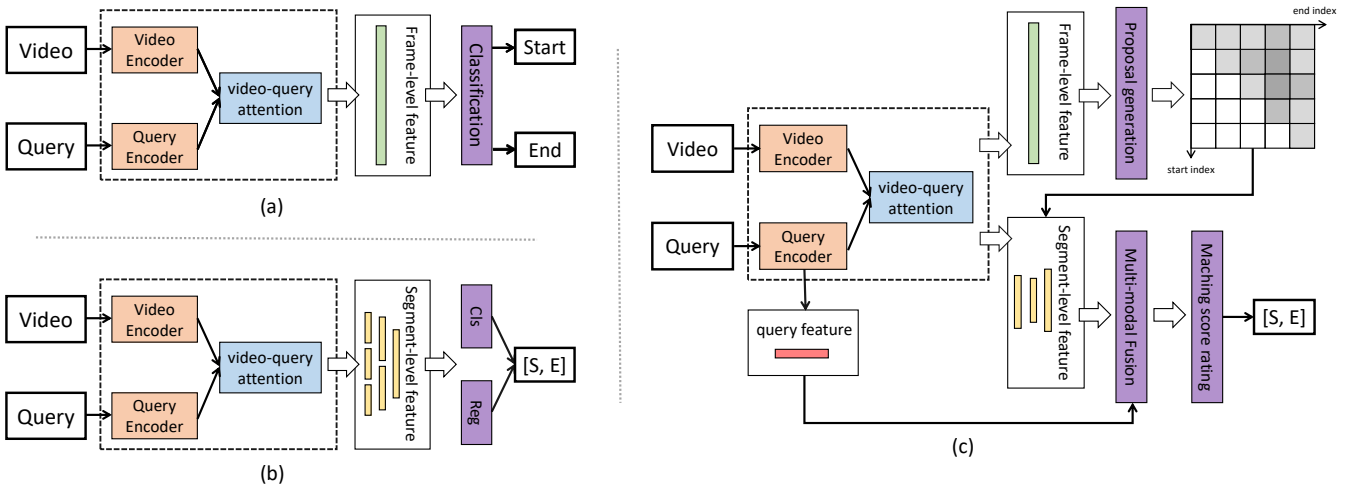


Figure 2: (a) A standard framework of anchor-free approach which conducts classification on frame-level visual feature. (b) A standard framework of anchor-based approach which does classification and regression on segment-level feature. (c) Architecture of the proposed two-stage framework: it first utilizes an anchor-free method to generate segment candidates and then computes a matching score with the query, which exploits both frame-level and segment-level information.

Huang, and Wang 2019) solving the NLVL task by reinforcement learning, which formulates the selection of start and end time points as a sequential decision making process.

**Anchor-based and Anchor-free Object Detection.** The development of NLVL is inspired by the success of object detection approaches. Object detection aims to obtain a tight bounding box and a class label for each object. It can be categorized into anchor-based and anchor-free approaches according to the way to localize an object. Traditional anchor-based models (Ren et al. 2015; Dai et al. 2016) have dominated this area for many years, which place a series of anchors (bounding boxes) uniformly and do the classification and regression to determine the position and class for the objects. Recently, researches on anchor-free models (Law and Deng 2018; Duan et al. 2019) are becoming prosperous, which have been promoted by the development of keypoint detection. The anchor-free methods directly predict the keypoints and group them together to determine the object. By the comparison of the two approaches, anchor-free methods are more flexible to locate objects with arbitrary geometry but have lower performance in contrast to the anchor-based methods because of the misalignment of keypoints. Duan et al. (2020) presented an anchor-free two-stage object detection framework termed CPN that extracts the keypoints and composes them into object proposals, then two-step classification is used to filter out the false positives. The BPNet borrows the similar idea from CPN which inherits the merits of both anchor-free and anchor-based approaches.

## Proposed Approach

We define the NLVL task as follows. Given an untrimmed video as $V = \{f_t\}_{t=1}^{T}$ and a language query as $Q = \{w_n\}_{m=1}^{M}$, where $T$ and $M$ are the number of video frames and query words, NLVL needs to predict the start time $t_s$ and the end time $t_e$ of the video segment described by the language query $Q$. For each video, we extract its visual features $\boldsymbol{V} = \{\boldsymbol{v}_t\}_{t=1}^{T}$ by a pre-trained 3D ConvNet (Tran et al. 2015). For each query, we initialize the word features $\boldsymbol{Q} = \{\boldsymbol{w}_n\}_{m=1}^{M}$ using the GloVe embeddings (Pennington, Socher, and Manning 2014).

As shown in Figure 2(c), our framework operates in two steps: boundary proposal generation and visual-language matching. Specifically, in the first step, it uses an anchor-free method to extract video segment proposals. In the second step, it fuses each segment proposal with the language and computes a matching score. The proposal with the highest matching score will be chosen as the correct segment.

In this section, we introduce the architecture of our BPNet. In Section , we first describe the boundary proposal generation phase. In Section , we then present the visual-language matching phase. Finally, in Section , we show the training and inference processes of BPNet in detail.

### Boundary Proposal Generation

The first stage is an anchor-free proposal extraction process, in which we generate a series of video segment proposals. Different from the existing anchor-based approaches, our generated proposals are of high quality because we utilize an effective anchor-free approach. We follow the anchor-free backbone of Video Span Localization Network (VSLNet) (Zhang et al. 2020a), which addressed the NLVL task as a span-based QA task. It is worth noting that our proposed BPNet is a universal framework that can incorporate any other anchor-free approach.

The first stage of BPNet consists of three components:

**Embedding Encoder Layer.** We use a similar encoder in QANet (Yu et al. 2018). The input of this layer is visual features $\boldsymbol{V} \in \mathbb{R}^{T \times d_v}$ and text query feature $\boldsymbol{Q} \in \mathbb{R}^{M \times d_q}$. We project them into the same dimension and feed them into the embedding encoder layer to integrate contextual infor-

mation.

$$\begin{aligned} \boldsymbol{V}^{'} &= \text{EmbeddingEncoder}(\boldsymbol{V}\boldsymbol{W}_v), \\ \boldsymbol{Q}^{'} &= \text{EmbeddingEncoder}(\boldsymbol{Q}\boldsymbol{W}_q), \end{aligned} \tag{1}$$

where $\boldsymbol{W}_v \in \mathbb{R}^{d_v \times d}$, $\boldsymbol{W}_q \in \mathbb{R}^{d_q \times d}$ are project matrices. Notice that the biases for transformation layers are omitted for clarity (the same below). As shown in Figure 3, the embedding encoder layer consists of multiple components, including four convolution layers, multi-head attention layer, layer normalization layer and feed-forward layer with the residual connection. The output of the embedding encoder layer $\boldsymbol{V}^{'} \in \mathbb{R}^{T \times d}$ and $\boldsymbol{Q}^{'} \in \mathbb{R}^{M \times d}$ are refined visual and language features that encode the interaction inside each modality.

**Visual-Language Attention Layer.** This layer calculates vision-to-language attention and language-to-vision attention weights and encodes the two modal features together. Specifically, if first computes a similarity matrix $\mathcal{S} \in \mathbb{R}^{T \times M}$, where the element $\mathcal{S}_{ij}$ indicates the similarity between the frame $f_i$ and the word $w_j$. Then the two attention weights $\boldsymbol{A}$ and $\boldsymbol{B}$ are computed:

$$\boldsymbol{A} = \mathcal{S}_{row} \cdot \boldsymbol{Q}^{'}, \quad \boldsymbol{B} = \mathcal{S}_{row} \cdot \mathcal{S}_{col}^{T} \cdot \boldsymbol{V}^{'}, \tag{2}$$

where $\mathcal{S}_{row}$ and $\mathcal{S}_{col}$ are the row and column-wise normalization of $\mathcal{S}$. We then model the interaction between the video and the query by the cross-modal attention layer:

$$\boldsymbol{V}^{q} = \text{FFN}([\boldsymbol{V}^{'}; \boldsymbol{A}; \boldsymbol{V}^{'} \odot \boldsymbol{A}; \boldsymbol{V}^{'} \odot \boldsymbol{B}]), \tag{3}$$

where $\odot$ is the element-wise multiplication, and $[\cdot]$ is the concatenation operation. The FFN represents feed-forward layer. The output of this layer $\boldsymbol{V}^{q}$ encodes the visual feature with query-guided attention.

**Proposal Generation Layer.** After getting the query-guided visual feature $\boldsymbol{V}^{q}$, we now generate proposals by using two stacked LSTMs, the hidden states of which are fed into two feed-forward layers to compute the start and end scores:

$$\boldsymbol{H}^{s} = \text{LSTM}_s(\boldsymbol{V}^{q}), \quad \boldsymbol{S}^{s} = \text{FFN}(\boldsymbol{H}^{s}), \tag{4}$$

$$\boldsymbol{H}^{e} = \text{LSTM}_e(\boldsymbol{H}^{s}), \quad \boldsymbol{S}^{e} = \text{FFN}(\boldsymbol{H}^{e}), \tag{5}$$

where $\boldsymbol{H}^{s}$ and $\boldsymbol{H}^{e}$ are the hidden states of the $\text{LSTM}_s$ and $\text{LSTM}_e$; $\boldsymbol{S}^{s}$ and $\boldsymbol{S}^{e}$ denote the logits of start and end boundaries computed by a feed-forward layer.

Then, we compute the joint probability of start and end points using matrix multiplication:

$$\begin{aligned} \boldsymbol{P}_s &= \text{softmax}(\boldsymbol{S}^{s}), \\ \boldsymbol{P}_e &= \text{softmax}(\boldsymbol{S}^{e}), \\ \boldsymbol{M}^{p} &= \boldsymbol{P}_s^{T} \boldsymbol{P}_e, \end{aligned} \tag{6}$$

where $\boldsymbol{P}_s$ and $\boldsymbol{P}_e$ are probability distributions of the start and end boundaries. $\boldsymbol{M}^{p}$ is a two-dimensional score map whose element indicates the predicted probability of each video segment, *e.g.*, $\boldsymbol{M}^{p}_{ij}$ denotes the score for the segment from start boundary $i$ to end boundary $j$. We sample the $N$ highest position on the score map $\boldsymbol{M}^{p}$ and treat the corresponding segments as candidates.

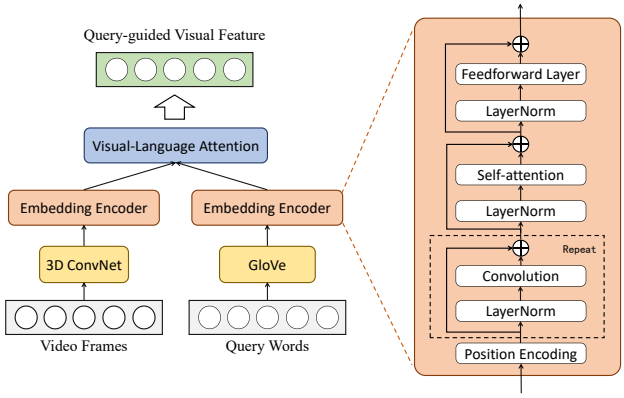


Figure 3: The backbone of BPNet (a variant of the QANet). The embedding encoder consists of several conv-layers, a self-attention layer and a feed-forward layer. For each layer, layer normalization and residual connection are employed.

## Visual-Language Matching

**Visual-Language Fusion Layer.** Given $N$ video segment candidates for video $V$, we capture the candidate features $\boldsymbol{C}$ from the visual feature $\boldsymbol{V}^{'}$ in Eq. (1). The generated video segment candidates have different lengths in the temporal dimension, hence we transform the candidate features into identical length using the temporal weighted pooling. We also obtain sentence-level query feature by weighted pooling over the word-level features. Then, we fuse them by concatenation followed by a feed-forward layer.

$$\begin{aligned} \widetilde{\boldsymbol{C}}_i &= \text{AvgPooling}(\boldsymbol{W}_c \boldsymbol{C}_i), \\ \widetilde{\boldsymbol{Q}} &= \text{AvgPooling}(\boldsymbol{W}_q \boldsymbol{Q}^{'}), \\ \boldsymbol{F} &= \text{FFN}([\widetilde{\boldsymbol{C}}_i, \widetilde{\boldsymbol{Q}}]), \end{aligned} \tag{7}$$

where $\boldsymbol{W}_c$ and $\boldsymbol{W}_q$ are learnable weights and $[\cdot]$ is the concatenation operation. This layer is to encode the segment-level information of video and fuse the visual and language feature for the subsequent process.

**Matching Score Rating Layer.** Taking the multi-modal feature as input, this layer predicts the matching score for each video segment proposal and the language query. The most matched proposal will be chosen as the final result. This layer consists of two feed-forward layers followed by ReLU and sigmoid activation respectively:

$$\hat{s}_i = \text{sigmoid}(\boldsymbol{W}_2 \text{ReLU}(\boldsymbol{W}_1 \boldsymbol{F})), \tag{8}$$

where $\hat{s}_i$ indicates the predicted matching score of the $i$-th candidates. We argue that the matching scores are positively associated with the temporal IoU scores between candidates and the ground-truth video segment. Therefore, we use the IoU scores as the ground-truth labels to supervise the training process. As a result, the matching score rating problem turns into an IoU regression problem.

## Training and Inference

**Training.** Each training sample consists of an untrimmed video, a language query and the ground-truth video segment.

Specifically, for each video frame with the frame-level feature, two class labels indicated whether or not the frame is the start or the end boundary are assigned. For each segment candidate with the segment-level feature, we compute the temporal IoU between the candidate and the ground-truth segment as the matching score.

There are two loss functions for the boundary proposal generation stage and visual-language matching stage:

**Boundary Classification Loss:**

$$\mathcal{L}_{cls} = f_{CE}(P_s, Y_s) + f_{CE}(P_e, Y_e), \tag{9}$$

where the $f_{CE}$ is a binary cross entropy loss function. $Y_s$ and $Y_e$ are ground-truth labels for the start and end boundaries.

**Matching Regression Loss:**

$$\mathcal{L}_{reg} = f_{MSE}(\hat{s}, s_{IoU}), \tag{10}$$

where $f_{MSE}$ is a $L2$ loss function and $s_{IoU}$ is the ground-truth temporal IoU scores.

Thus, the final loss is a multi-task loss combining the $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$, *i.e.*,

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \times \mathcal{L}_{reg}, \tag{11}$$

where $\lambda$ is a hyper-parameter that balances the two losses.

**Inference.** Given a video and a language query, we forward them through the network and obtain $N$ segment candidates with their corresponding matching scores $\hat{s}$. Then, we rank the $\hat{s}$ and select the candidate with the highest score as the final result.

# Experiments

## Datasets

We evaluate our BPNet on three public benchmark datasets: 1) **Charades-STA** (Gao et al. 2017): It is built on Charades and contains 6,672 videos of daily indoors activities. Charades-STA contains 16,128 sentence-moment pairs in total, where 12,408 pairs are for training and 3,720 pairs for testing. The average duration of the videos is 30.59s and the average duration of the video segments is 8.22s. 2) **TACoS**: It consists of 127 videos of cooking activities. For video grounding task, it contains 18818 sentence-moment pairs in total. Followed by the split setting in (Gao et al. 2017), we use 10,146, 4,589, 4,083 for training, validation and testing respectively. The duration of the videos is 287.14s on average and the average length of the video segments is 5.45s. 3) **ActivityNet Captions** (Krishna et al. 2017): It contains around 20k open domain videos for video grounding task. We follow the split in (Yuan, Mei, and Zhu 2019), which consists of 37,421 sentence-moment pairs for training and 17,505 for testing. The average duration of the videos is 117.61s and the average length of the video segments is 36.18s.

## Evaluation Metrics

Following the prior works, we adopt "R@$n$, IoU=$\theta$" and "mIoU" as evaluation metrics. Specifically,"R@$n$, IoU=$\theta$" represents the percentage of the testing samples that have at least one of the top-N results whose IoU with the ground-truth is larger than $\theta$. The "mIoU" means the average IoU with ground truth over all testing samples. In all the experiments, we set $n$ = 1 and $\theta \in \{0.3, 0.5, 0.7\}$.

## Implementation

We down-sample frames for each video and extract visual features using C3D (Tran et al. 2015) network pretrained on Sports-1M. Then we reduce the features to 500 dimension by PCA. For language query, we initialize each word with 300d GloVe vectors and all word embeddings are fixed during training. The dimension of the intermediate layer in BPNet is set to 128. The number of convolution blocks in embedding encoder is 4 and the kernel size is set to 7. The number of boundary proposals is 128 for training and 8 for testing. For all datasets, we trained the model for 100 epochs with batch size of 32. Dropout and early stopping strategies are adopted to prevent overfitting. We implement our BPNet on Tensorflow. The whole framework is trained by Adam optimizer with learning rate 0.0001.

## Comparisons with the State-of-the-Arts

**Settings.** We compare the proposed BPNet with several state-of-the-art NLVL methods on three datasets. These methods are grouped into three categories by the viewpoints of anchor-based and anchor-free approach: 1) Anchor-based models: **VSA_RNN**, **VSA_STV**, **CTRL** (Gao et al. 2017), **ACRN** (Liu et al. 2018a), **ROLE** (Liu et al. 2018b), **MCF** (Wu and Han 2018), **ACL** (Ge et al. 2019), **SAP** (Chen and Jiang 2019), **QSPN** (Xu et al. 2019), **TGN** (Chen et al. 2018), **MAN** (Zhang et al. 2019). 2) Anchor-free models: **L-Net** (Chen et al. 2019), **ABLR-af**, **ABLR-aw** (Yuan, Mei, and Zhu 2019), **DEBUG** (Lu et al. 2019), **ExCL** (Ghosh et al. 2019), **GDP** (Chen et al. 2020), **VSLNet** (Zhang et al. 2020a). 3) Others: **RWM** (He et al. 2019), **SM-RL** (Wang, Huang, and Wang 2019).

The results on three benchmarks are reported in Table 1 to Table 3. We can observe that our BPNet achieves new state-of-the-art performance over all metrics and benchmarks. Table 1 summarizes the results on Charades-STA. We can observe that BPNet outperforms all the baselines in all metrics. Specifically, we observe that BPNet works well in even stricter metrics, *e.g.*, BPNet achieved a significant 2.59 absolute improvement in IoU@0.7 compared to the second result, which demonstrates the effectiveness of our model. For a fair comparison with VSLNet (Zhang et al. 2020a), we use both C3D and I3D (Carreira and Zisserman 2017) visual features. VSLNet is a typical anchor-free model with state-of-the-art performance whose architecture is roughly described in Figure 2 (a). Specifically, we implement the VSLNet with C3D feature followed the settings they reported. VSLNet extracts the frame-level feature using the backbone of QANet and utilizes two LSTM to classify the start/end boundary. Our model outperforms VSLNet in all metrics on Charades-STA. It is mainly because that BPNet better models the segment-level visual information between the boundaries.

The results on TACoS and ActivityNet Captions are summarized in Table 2 and Table 3. Note that the videos in TACoS have a longer average duration and the ground-truth video segments in ActivityNet Captions have a longer average length. BPNet significantly outperforms the other methods on both benchmarks with the C3D feature, which

| Methods | Feature | IoU=0.3 | IoU=0.5 | IoU=0.7 | mIoU |
|---|---|---|---|---|---|
| VSA-RNN | C3D | – | 10.50 | 4.32 | – |
| VSA-STV | C3D | – | 16.91 | 5.81 | – |
| CTRL | C3D | – | 23.63 | 8.89 | – |
| ROLE | C3D | 25.26 | 12.12 | – | – |
| ACL-K | C3D | – | 30.48 | 12.20 | – |
| SAP | C3D | – | 27.42 | 13.36 | – |
| RWM | C3D | – | 36.70 | – | – |
| SM-RL | C3D | – | 24.36 | 11.17 | – |
| QSPN | C3D | 54.70 | 35.60 | 15.80 | – |
| DEBUG | C3D | 54.95 | 37.39 | 17.92 | 36.34 |
| GDP | C3D | 54.54 | 39.47 | 18.49 | – |
| VSLNet | C3D | 54.38 | 28.71 | 15.11 | 37.07 |
| BPNet | C3D | **55.46** | **38.25** | **20.51** | **38.03** |
| ExCL | I3D | – | 44.10 | 22.40 | – |
| MAN | I3D | – | 46.53 | 22.72 | – |
| VSLNet | I3D | 64.30 | 47.31 | 30.19 | 45.15 |
| BPNet | I3D | **65.48** | **50.75** | **31.64** | **46.34** |

Table 1: Performance (%) of "R@$n$, IoU=$\theta$" and "mIoU" compared with the state-of-the-art NLVL models on Charades-STA.

| Methods | IoU=0.3 | IoU=0.5 | IoU=0.7 | mIoU |
|---|---|---|---|---|
| VSA-RNN | 6.91 | – | – | – |
| VSA-STV | 10.77 | – | – | – |
| CTRL | 18.32 | 13.30 | – | – |
| ACRN | 19.52 | 14.62 | – | – |
| MCF | 18.64 | – | – | – |
| SM-RL | 20.25 | 15.95 | – | – |
| ACL | 22.07 | 17.78 | – | – |
| SAP | – | 18.24 | – | – |
| L-NET | – | – | – | 13.41 |
| TGN | 21.77 | 18.90 | – | – |
| ABLR-aw | 18.90 | 9.30 | – | 12.50 |
| ABLR-af | 19.50 | – | – | 13.40 |
| DEBUG | 23.45 | 11.72 | – | 16.03 |
| GDP | 24.14 | – | – | 16.18 |
| VSLNet | 22.88 | 18.98 | 14.01 | 18.01 |
| BPNet | **25.96** | **20.96** | **14.08** | **19.53** |

Table 2: Performance (%) of "R@$n$, IoU=$\theta$" and "mIoU" compared with the state-of-the-art NLVL models on TACoS.

demonstrates that BPNet is highly adaptive to videos and segments with diverse lengths. The qualitative results of BPNet is illustrated in Figure 4.

It is worth noting that BPNet can utilize a more effective anchor-free backbone such as (Zhang et al. 2020b; Zeng et al. 2020) to further improve the performance. Even so, we take into account the simplicity and efficiency and choose the VSLNet as the backbone.

## Ablation Study

In this section, we conduct ablative experiments with different variants to better investigate our approach.

**Anchor-based vs. Anchor-free.** To evaluate the effectiveness of our two-stage model, we compare BPNet with both

| Methods | IoU=0.3 | IoU=0.5 | IoU=0.7 | mIoU |
|---|---|---|---|---|
| TGN | 43.81 | 27.93 | – | – |
| QSPN | 45.30 | 27.70 | 13.60 | – |
| RWM | – | 36.90 | – | – |
| ABLR-af | 53.65 | 34.91 | – | 35.72 |
| ABLR-aw | 55.67 | 36.79 | – | 36.99 |
| DEBUG | 55.91 | 39.72 | – | 39.51 |
| GDP | 56.17 | 39.27 | – | 18.49 |
| VSLNet | 55.17 | 38.34 | 23.52 | 40.53 |
| BPNet | **58.98** | **42.07** | **24.69** | **42.11** |

Table 3: Performance (%) of "R@$n$, IoU=$\theta$" and "mIoU" compared with the state-of-the-art NLVL models on ActivityNet Captions.

| Charades-STA | | | | |
|---|---|---|---|---|
| Methods | IoU=0.3 | IoU=0.5 | IoU=0.7 | mIoU |
| VSLNet (anchor-free) | 54.38 | 28.71 | 15.11 | 37.07 |
| Our (anchor-based) | 55.97 | 34.81 | 15.46 | 35.94 |
| BPNet | **55.46** | **38.26** | **20.51** | **38.03** |
| TACoS | | | | |
| Methods | IoU=0.3 | IoU=0.5 | IoU=0.7 | mIoU |
| VSLNet (anchor-free) | 22.88 | 18.98 | 14.01 | 18.01 |
| Our (anchor-based) | 22.39 | 14.67 | 7.42 | 15.63 |
| BPNet | **25.96** | **20.96** | **14.08** | **19.53** |
| ActivityNet Captions | | | | |
| Methods | IoU=0.3 | IoU=0.5 | IoU=0.7 | mIoU |
| VSLNet (anchor-free) | 55.17 | 38.34 | 23.52 | 40.53 |
| Our (anchor-based) | 58.75 | 40.52 | 18.74 | 39.66 |
| BPNet | **58.98** | **42.07** | **24.69** | **42.11** |

Table 4: Performance (%) comparisons of the anchor-free model, anchor-based model and BPNet with the same backbone on three benchmarks.

anchor-free and anchor-based models. For a fair comparison, we designed an anchor-based model with the same backbone as BPNet. Specifically, we sampled a series of bounding boxes over the temporal dimension and conduct the following matching process. Since VSLNet has the same backbone as BPNet, we use it as the anchor-free setting.

*Results.* The results of the three models are reported in Table 4. We can observe that our BPNet outperforms all the baselines with the same backbone. In particular, we observe that our implementation of the anchor-based model works well on looser metrics (*e.g.*, anchor-based) while the anchor-free model does better on strict metrics. We think that's because the anchor-based model has wider coverage bounding boxes. BPNet performed better than both models over all metrics.

**Quality of the Candidates.** We compare the quality of the candidates generated by our BPNet and by dense sampling bounding boxes. The results conducted on TACoS are reported in Table 5. In the anchor-based setting, we perform bounding boxes uniformly on video frames. The lengths of sliding windows are 8, 16, 32, 64 and 128, window's overlap is 0.75. We evaluate the anchor-based model with different window lengths, which results in variant number of

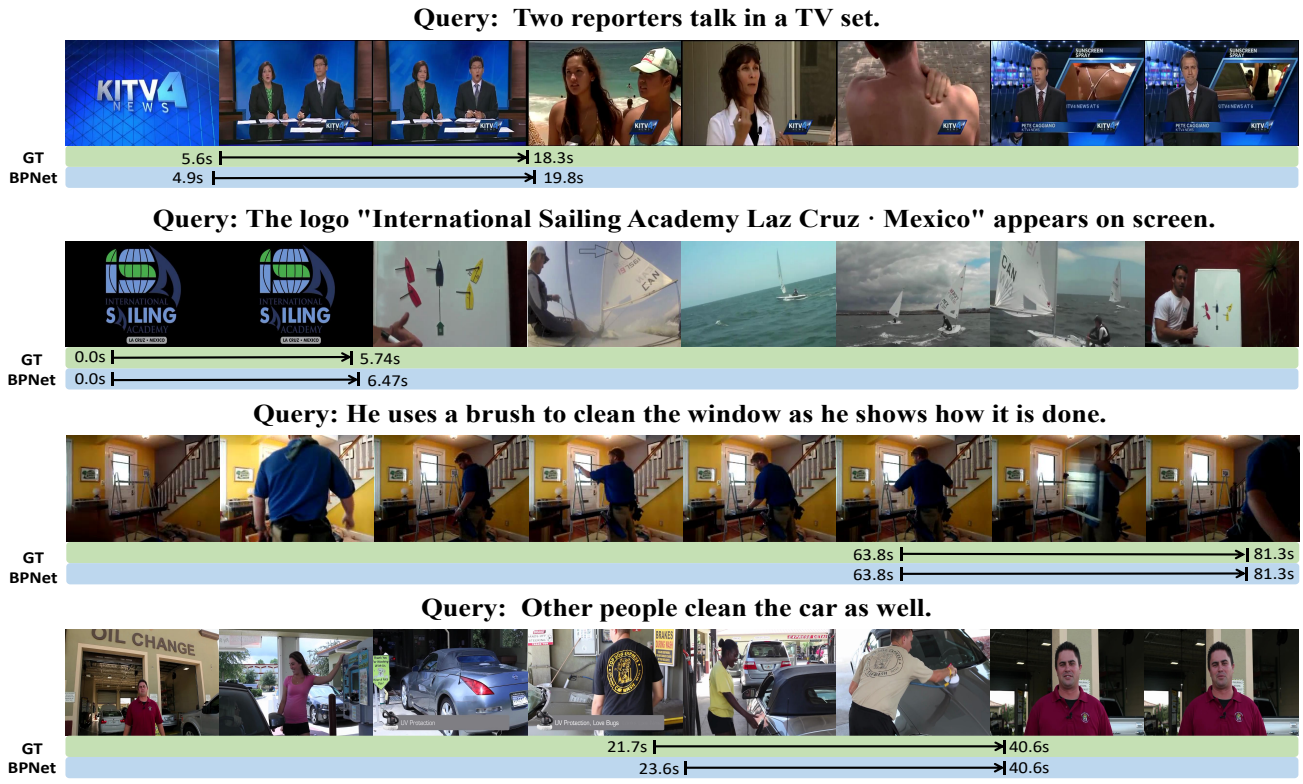


Figure 4: The qualitative results of BPNet on ActivityNet Captions.

| Methods | Length of Windows | | | | | num | mIoU |
|---|---|---|---|---|---|---|---|
| | 128 | 64 | 32 | 16 | 8 | | |
| Anchor-based Model | ✓ | | | | | 5 | 3.92 |
| | | ✓ | | | | 13 | 5.88 |
| | | | ✓ | | | 29 | 8.62 |
| | | | | ✓ | | 61 | 12.32 |
| | | | | | ✓ | 125 | 13.50 |
| | ✓ | ✓ | ✓ | ✓ | ✓ | 233 | 15.65 |
| BPNet | no limit | | | | | 8 | **19.53** |

Table 5: Performance (%) comparisons with the anchor-based model in different settings on TACoS.

candidates. We notice that our BPNet only generated 8 candidates which are much less than the anchor-based setting but achieved a higher performance. This indicates that BPNet can generate high-quality candidates.

**With vs. Without Visual-Language Fusion Layer.** We evaluate the model without multi-modal fusion before matching score rating. For a fair comparison, we use the multi-modal features $V^q$ in Eq. (3) for this setting. From Table 6, we can observe that the Visual-Language Fusion Layer improves the performance. The main reason is that the multi-modal fusion layer is able to model the segment-level video-query interaction.

| Method | IoU=0.3 | IoU=0.5 | IoU=0.7 | mIoU |
|---|---|---|---|---|
| BPNet w/o vlf | 55.05 | 34.78 | 18.84 | 37.92 |
| BPNet | **55.46** | **38.25** | **20.51** | **38.03** |

Table 6: Performance (%) comparisons on Charades-STA. BPNet w/o vlf represent the BPNet without the Visual-Language Fusion layer.

## Conclusions

In this paper, we propose a novel Boundary Proposal Network (BPNet) for natural language video localization (NLVL). By utilizing an anchor-free model to generate high-quality video segment candidates, we disentangle the candidate proposals from the predefined heuristic rules to make them adaptable to video segments with variant lengths. Furthermore, we jointly model the segment-level video feature and query feature, which further boosts the performance. As a result, the proposed BPNet outperforms the state-of-the-art approaches on three benchmark datasets. Moreover, BPNet is a universal framework which means that the proposal generation module and the visual-language matching module can be replaced by any other effective methods. In the future, we are going to extend this framework into other related tasks, *e.g.*, visual grounding (Chen et al. 2021).

## Acknowledgments

This work was supported by the National Key Research & Development Project of China (2018AAA0101900), the National Natural Science Foundation of China (U19B2043, 61976185), Zhejiang Natural Science Foundation (LR19F020002, LZ17F020001), Key Research & Development Project of Zhejiang Province(2018C03055), Major Project of Zhejiang Social Science Foundation (21XXJC01ZD), and the Fundamental Research Funds for the Central Universities.

## References

Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 4724–4733.

Chen, J.; Chen, X.; Ma, L.; Jie, Z.; and Chua, T. 2018. Temporally Grounding Natural Sentence in Video. In *EMNLP*, 162–171.

Chen, J.; Ma, L.; Chen, X.; Jie, Z.; and Luo, J. 2019. Localizing Natural Language in Videos. In *AAAI*, 8175–8182.

Chen, L.; Lu, C.; Tang, S.; Xiao, J.; Zhang, D.; Tan, C.; and Li, X. 2020. Rethinking the Bottom-Up Framework for Query-Based Video Localization. In *AAAI*, 10551–10558.

Chen, L.; Ma, W.; Xiao, J.; Zhang, H.; and Chang, S.-F. 2021. Ref-NMS: Breaking Proposal Bottlenecks in Two-Stage Referring Expression Grounding. In *AAAI*.

Chen, S.; and Jiang, Y. 2019. Semantic Proposal for Activity Localization in Videos via Sentence Query. In *AAAI*, 8199–8206.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *NIPS*, 379–387.

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. CenterNet: Keypoint Triplets for Object Detection. In *ICCV*, 6568–6577.

Duan, K.; Xie, L.; Qi, H.; Bai, S.; Huang, Q.; and Tian, Q. 2020. Corner Proposal Network for Anchor-free, Two-stage Object Detection. In *ECCV*, 399–416.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. TALL: Temporal Activity Localization via Language Query. In *ICCV*, 5277–5285.

Ge, R.; Gao, J.; Chen, K.; and Nevatia, R. 2019. MAC: Mining Activity Concepts for Language-Based Temporal Localization. In *WACV*, 245–253.

Ghosh, S.; Agarwal, A.; Parekh, Z.; and Hauptmann, A. G. 2019. ExCL: Extractive Clip Localization Using Natural Language Descriptions. In *NAACL*, 1984–1990.

He, D.; Zhao, X.; Huang, J.; Li, F.; Liu, X.; and Wen, S. 2019. Read, Watch, and Move: Reinforcement Learning for Temporally Grounding Natural Language Descriptions in Videos. In *AAAI*, 334–343.

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. C. 2017. Localizing Moments in Video with Natural Language. In *ICCV*, 5804–5813.

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. C. 2018. Localizing Moments in Video with Temporal Language. In *EMNLP*, 1380–1390.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Niebles, J. C. 2017. Dense-Captioning Events in Videos. In *ICCV*, 706–715.

Law, H.; and Deng, J. 2018. CornerNet: Detecting Objects as Paired Keypoints. In *ECCV*, 765–781.

Lei, J.; Yu, L.; Bansal, M.; and Berg, T. L. 2018. TVQA: Localized, Compositional Video Question Answering. In *EMNLP*, 1369–1379.

Liu, M.; Wang, X.; Nie, L.; He, X.; Chen, B.; and Chua, T. 2018a. Attentive Moment Retrieval in Videos. In *SIGIR*, 15–24.

Liu, M.; Wang, X.; Nie, L.; Tian, Q.; Chen, B.; and Chua, T. 2018b. Cross-modal Moment Localization in Videos. In *MM*, 843–851.

Lu, C.; Chen, L.; Tan, C.; Li, X.; and Xiao, J. 2019. DEBUG: A Dense Bottom-Up Grounding Approach for Natural Language Video Localization. In *EMNLP*, 5143–5152.

Mun, J.; Cho, M.; and Han, B. 2020. Local-Global Video-Text Interactions for Temporal Grounding. In *CVPR*, 10807–10816.

Opazo, C. R.; Marrese-Taylor, E.; Saleh, F. S.; Li, H.; and Gould, S. 2020. Proposal-free Temporal Moment Localization of a Natural-Language Query in Video using Guided Attention. In *WACV*, 2453–2462.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*, 1532–1543.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 91–99.

Shao, D.; Xiong, Y.; Zhao, Y.; Huang, Q.; Qiao, Y.; and Lin, D. 2018. Find and Focus: Retrieve and Localize Video Events with Natural Language Queries. In *ECCV*, 202–218.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*, 4489–4497.

Wang, J.; Ma, L.; and Jiang, W. 2020. Temporally Grounding Language Queries in Videos by Contextual Boundary-Aware Prediction. In *AAAI*, 12168–12175.

Wang, W.; Huang, Y.; and Wang, L. 2019. Language-Driven Temporal Activity Localization: A Semantic Matching Reinforcement Learning Model. In *CVPR*, 334–343.

Wu, A.; and Han, Y. 2018. Multi-modal Circulant Fusion for Video-to-Language and Backward. In *IJCAI*, 1029–1035.

Xiao, S.; Li, Y.; Ye, Y.; Chen, L.; Pu, S.; Zhao, Z.; Shao, J.; and Xiao, J. 2020. Hierarchical Temporal Fusion of Multi-grained Attention Features for Video Question Answering. *Neural Processing Letters* 993–1003.

Xu, H.; He, K.; Plummer, B. A.; Sigal, L.; Sclaroff, S.; and Saenko, K. 2019. Multilevel Language and Vision Integration for Text-to-Clip Retrieval. In *AAAI*, 9062–9069.

Ye, Y.; Zhao, Z.; Li, Y.; Chen, L.; Xiao, J.; and Zhuang, Y. 2017. Video question answering via attribute-augmented attention network learning. In *SIGIR*, 829–832.

Yu, A. W.; Dohan, D.; Luong, M.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *ICLR*.

Yuan, Y.; Mei, T.; and Zhu, W. 2019. To Find Where You Talk: Temporal Sentence Localization in Video with Attention Based Location Regression. In *AAAI*, 9159–9166.

Zeng, R.; Xu, H.; Huang, W.; Chen, P.; Tan, M.; and Gan, C. 2020. Dense Regression Network for Video Grounding. In *CVPR*, 10284–10293.

Zhang, D.; Dai, X.; Wang, X.; Wang, Y.; and Davis, L. S. 2019. MAN: Moment Alignment Network for Natural Language Moment Retrieval via Iterative Graph Adjustment. In *CVPR*, 1247–1257.

Zhang, H.; Sun, A.; Jing, W.; and Zhou, J. T. 2020a. Span-based Localizing Network for Natural Language Video Localization. In *ACL*, 6543–6554.

Zhang, S.; Peng, H.; Fu, J.; and Luo, J. 2020b. Learning 2D Temporal Adjacent Networks for Moment Localization with Natural Language. In *AAAI*, 12870–12877.