

# Natural Language Video Localization with Learnable Moment Proposals

Shaoning Xiao<sup>†</sup>, Long Chen<sup>‡\*</sup>, Jian Shao<sup>†</sup>, Yueting Zhuang<sup>†</sup>, and Jun Xiao<sup>†</sup>

<sup>†</sup>DCD Lab, College of Computer Science, Zhejiang University

<sup>‡</sup>DVMM Lab, Columbia University

shaoningx@zju.edu.cn, zjuchenlong@gmail.com

jshao@zju.edu.cn, yzhuang@zju.edu.cn, junx@zju.edu.cn

## Abstract

Given an untrimmed video and a natural language query, Natural Language Video Localization (NLVL) aims to identify the video moment described by the query. To address this task, existing methods can be roughly grouped into two groups: 1) *propose-and-rank* models first define a set of hand-designed moment candidates and then find out the best-matching one. 2) *proposal-free* models directly predict two temporal boundaries of the referential moment from frames. Currently, almost all the propose-and-rank methods have inferior performance than proposal-free counterparts. In this paper, we argue that propose-and-rank approach is underestimated due to the predefined manners: 1) Hand-designed rules are hard to guarantee the complete coverage of targeted segments. 2) Densely sampled candidate moments cause redundant computation and degrade the performance of ranking process. To this end, we propose a novel model termed **LP-Net** (Learnable Proposal Network for NLVL) with a fixed set of learnable moment proposals. The position and length of these proposals are dynamically adjusted during training process. Moreover, a boundary-aware loss has been proposed to leverage frame-level information and further improve the performance. Extensive ablations on two challenging NLVL benchmarks have demonstrated the effectiveness of LPNet over existing state-of-the-art methods<sup>1</sup>.

## 1 Introduction

Natural Language Video Localization (NLVL), *aka*, video grounding or video moment localization, has got unprecedented attention in both CV and NLP communities (Gao et al., 2017; Hendricks et al., 2017). As shown in Figure 1, NLVL aims to local-

**Language Query:** The lady spins the stick using her neck.

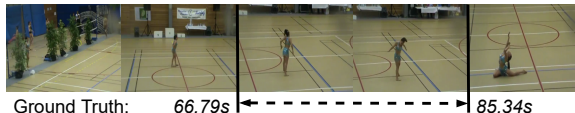


Figure 1: An illustrative example of NLVL. Given a video and a query, NLVL is to localize the video segment corresponding to the query with the start timestamp (66.79s) and the end timestamp (85.34s).

ize the video segment relevant to the query by locating the start and end timestamps in an untrimmed video. It is challenging since it needs to not only understand the video and the sentence content but also find out the precise temporal boundaries. Moreover, NLVL is helpful to numerous downstream video understanding tasks, *e.g.*, content retrieval (Shao et al., 2018), relation detection (Gao et al., 2021), and VQA (Lei et al., 2018; Ye et al., 2017).

Currently, state-of-the-art NLVL methods can be roughly grouped into two categories according to how the video segments are detected, namely *propose-and-rank* and *proposal-free* methods:

The idea of the *propose-and-rank* approach (Gao et al., 2017; Hendricks et al., 2018; Liu et al., 2018b; Chen et al., 2018; Ge et al., 2019; Xu et al., 2019; Zhang et al., 2019) is intuitive, which follows the same spirits of anchor-based object detectors, *e.g.*, Faster R-CNN (Ren et al., 2015). This kind of methods firstly defines a series of manually-designed temporal bounding boxes as moment proposals. Then, they match each candidate with the sentence in a common feature space and compute matching scores for all the candidates. Thus the localization problem is reformulated into a ranking problem. However, these methods suffer from two inherent drawbacks due to the predefined manners: 1) Even though they elaborately design a series of hyperparameters (*e.g.*, temporal scales and sample rates), these hand-designed rules are hard to guarantee the complete coverage of targeted video

<sup>\*</sup>Long Chen is the corresponding author. The work started when Long Chen was at Zhejiang University.

<sup>1</sup>Source codes is available: <https://github.com/xiaoneil/LPNet/>

segments, and consequently tend to produce inaccurate boundaries. 2) A vast number of proposals are required to achieve high recall, which causes redundant computation and degrades the results of ranking process.

Another type of solution *proposal-free* approach (Chen and Jiang, 2019; Yuan et al., 2019; Lu et al., 2019; Chen et al., 2020; Zhang et al., 2020a) mitigates these defects. Instead of predefining a series of temporal proposals, they directly predict the start and the end boundaries or regress the locations of the query-related video segments. Benefit from such design, proposal-free methods get rid of placing superfluous temporal anchors, *i.e.*, they are more computation-efficient. Furthermore, without fixing the position and length of the moment proposals, these methods are flexible to adapt to video segments with diverse lengths. Compared to propose-and-rank methods, there are two main limitations of proposal-free methods (Xiao et al., 2021): 1) They overlook the rich information between start and end boundaries because they are hard to model the segment-level interaction. 2) They always suffer from severe imbalance between the positive and negative training samples.

Up to now, almost all the propose-and-rank methods have inferior performance. We argue that the performance of the propose-and-rank methods are underestimated due to current predefined designs. In this paper, we propose a novel propose-and-rank model with learnable moment proposals, termed **LPNet**. Without fixed dense proposals, only a sparse set of proposals are required to obtain decent performance. In addition, there is no need to worry about the design of hyper-parameters because it is adaptable to targeted segments with diverse positions and lengths. Obviously, as a propose-and-rank method, LPNet also avoids the defects of the proposal-free approach.

Specifically, LPNet places a fixed set of learnable temporal proposals represented by 2-d coordinates indicating the centers and lengths of video segments. These proposals are used to extract visual features of Moment of Interest (MoI). In order to model the relative relations among candidates, a module has been proposed to make the candidates interact with each other using the self-attention mechanism. Then an individual classifier is used to predict the matching score between these proposals and the sentence query. During the training process, the coordinates of the proposal

with maximum score are adjusted by a dynamic adjuster at each iteration. After sufficient iterations, these learned moment proposals will statistically represent the prior distributions of ground-truth segments on the dataset. In addition, we empirically find that the propose-and-rank models always obtain sub-optimal results without frame-level supervision. A boundary-aware predictor has been proposed to regularize the model to utilize frame-level information, which further boosts the grounding performance.

We demonstrate the effectiveness of our model on two challenging NLVL benchmarks (Charades-STA, and ActivityNet Captions) by extensive ablation studies. Particularly, our model achieves new state-of-the-art performance over all datasets and evaluation metrics.

## 2 Related Work

**Natural Language Video Localization.** NLVL task was first introduced in (Hendricks et al., 2017; Gao et al., 2017). Current existing methods can be roughly grouped into two categories, namely *propose-and-rank* and *proposal-free* methods.

The propose-and-rank 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 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 MCN which effectively localizes language queries in videos by integrating local and global video features over time. To improve the performance of the propose-and-rank 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

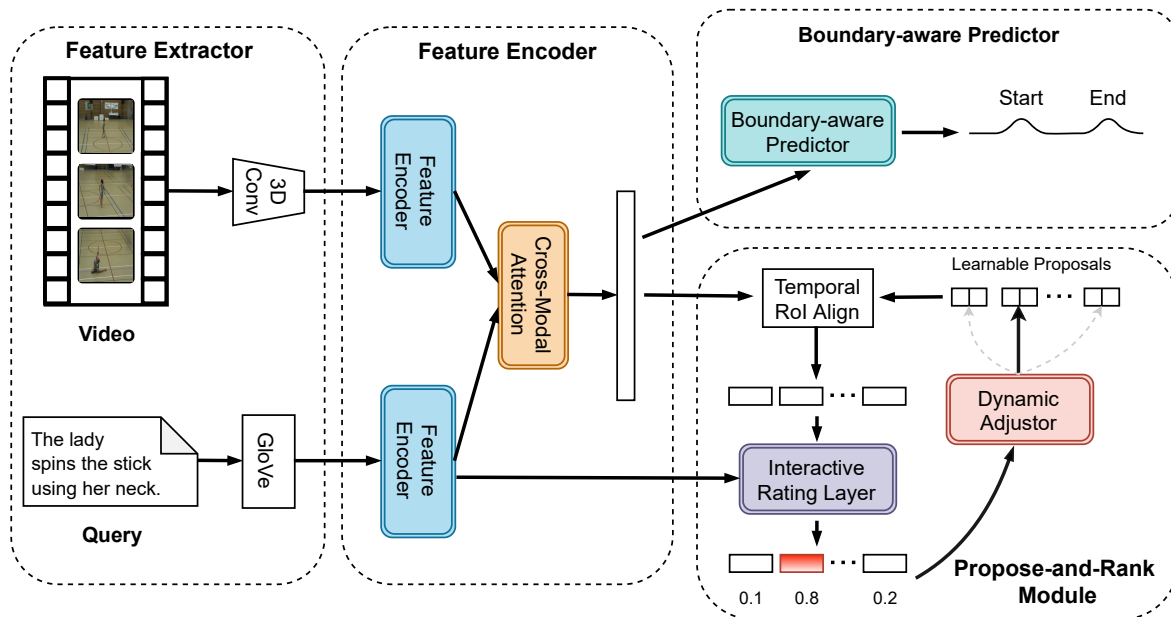


Figure 2: The architecture of LPNet for NLVL. Feature extractor transforms input video and language query into feature space. Feature encoder further refines video and language feature, and produces the multi-modal feature. A series of learnable proposal boxes are proposed which can be updated by dynamic adjustor during training. Interactive rating layer scores each candidate generated by proposal boxes and the candidate with highest score is the final prediction. Boundary-aware predictor takes multi-modal feature as input and predict the distribution of start and end timestamps, which is an auxiliary task to regularize the model to get better performance.

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.

Proposal-free approaches (Yuan et al., 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 or regress the location. Yuan et al. (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 classified 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 boundaries.

There are also some other works (He et al., 2019; Wang et al., 2019) solving the NLVL task by RL,

which formulates the selection of start and end timestamps as a sequential decision making process. And some concurrent NLVL works also borrow the design of Visual Transformers (Dosovitskiy et al., 2021; Wang et al., 2021) to explore Transformer-based NLVL model (Cao et al., 2021).

**End-to-End Object Detection.** The development of NLVL is inspired by the success of object detection methods. 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. Traditional anchor-based models (Ren et al., 2015; Dai et al., 2016) have dominated this area for many years, which place a series of anchor boxes uniformly and do the classification and regression to determine the position and class for the objects. Anchor-free models (Law and Deng, 2018; Duan et al., 2019) are becoming prosperous, which have been promoted by the development of key point detection. They directly predict key points and group them together to determine objects. Recently, end-to-end object detectors based on sparse candidates have drawn large amount of attentions. DETR (Carion et al., 2020) utilizes a sparse set of object queries to interact with the global feature. Benefit from the bipartite matching predictions and ground-truths, DETR can discard

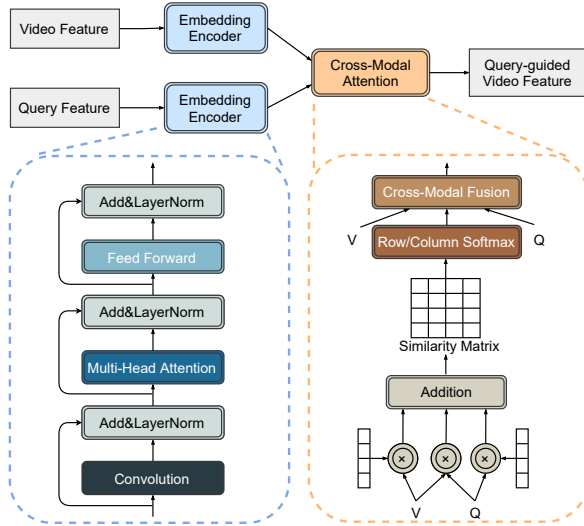


Figure 3: The overview of the Feature Encoder. The embedding encoder consists of several conv-layers, a self-attention layer and a feed-forward layer together with layer normalization and residual connection. The cross-modal attention layer learns the multi-modal interaction and outputs the query-guided video feature.

the NMS procedure while achieving remarkable performance. Similarly, Sparse R-CNN (Sun et al., 2021) provides a fixed set of sparse learnable object proposals and performs classification and localization. Our model borrows the similar idea from Sparse R-CNN.

### 3 Approach

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

The overall architecture of the proposed LPNet is shown in Figure 2. In this section, we first introduce each component of LPNet in sequence. Then, we show the training and inference stage in details.

#### 3.1 Feature Encoder

**Embedding Encoder Layer.** Following previous works (Lu et al., 2019), we use a similar feature encoder in QANet (Yu et al., 2018). The embedding encoder layer consists of multiple components as

shown in the left of Figure 3. The input of this layer is visual features  $V \in \mathbb{R}^{T \times d_v}$  and text features  $Q \in \mathbb{R}^{M \times d_q}$ . We project them into the same dimension and feed them into the embedding encoder layer respectively to integrate contextual information. The output of the embedding encoder layer  $V' \in \mathbb{R}^{T \times d}$  and  $Q' \in \mathbb{R}^{M \times d}$  is refined visual and text features that encode the interaction under each modality.

**Cross-Modal Attention Layer.** This layer calculates vision-to-language attention and language-to-vision attention weights, and then encodes the multi-modal feature. As shown in the right of Figure 3, it first computes a similarity matrix  $S \in \mathbb{R}^{T \times M}$ , where the element  $S_{ij}$  indicates the similarity between the frame  $f_i$  and the word  $w_j$ . Then, the two attention weights  $A$  and  $B$  are computed:

$$A = S_{row} \cdot Q', \quad B = S_{row} \cdot S_{col}^T \cdot V', \quad (1)$$

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

$$V^q = \text{FFN}([V'; A; V' \odot A; V' \odot B]), \quad (2)$$

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

#### 3.2 Propose-and-Rank Module

**Learnable Moment Proposals.** Different from the previous manually designed temporal anchors, our proposal boxes are learnable during the training process. We define the number of proposals as  $N$ . The proposal boxes are represented by 2-d parameters ranging from 0 to 1 which are randomly initialized, denoting normalized center coordinates and lengths. The parameters of proposal boxes ( $N \times 2$ ) will be dynamically adjusted with the back-propagation. In order to model the implicit relation among proposals, following (Sun et al., 2021), we attach *proposal features*  $P \in \mathbb{R}^{N \times d}$  to every proposal boxes. Simultaneously, multi-head self-attention (MHSA) mechanism is applied to proposal features to reason about interactions among proposals:  $\tilde{P} = \text{MHSA}(P)$ .

**Interactive Rating Layer.** Given  $N$  moment proposal boxes for video  $V$ , we capture the candidate features  $C$  from the visual feature  $V^q$  in Eq. (2).



The generated video segment candidates have different lengths in the temporal dimension, hence we transform the candidate features into identical length using temporal RoIAlign. For  $i$ -th candidate feature  $C_i$ :

$$C'_i = \text{RoIAlign}(C_i), \quad (3)$$

where  $C'_i \in \mathbb{R}^{l \times d}$ . We then interact the candidate feature with its corresponding proposal feature  $\tilde{P}_i \in \mathbb{R}^d$  to encode richer information following (Sun et al., 2021):

$$\tilde{C}_i = \text{Flatten}(C'_i W_p \tilde{P}_i) W_c, \quad (4)$$

where  $W_p \in \mathbb{R}^{d \times d}$  and  $W_c \in \mathbb{R}^{l \times d}$  are learnable weights, and Flatten is an operation that flattens the matrix to one-dimensional vector. We also obtain sentence-level query feature  $\tilde{Q} \in \mathbb{R}^d$  by weighted pooling over word-level features  $Q'$ . Then, we fuse them by concatenation followed by a feed-forward layer:

$$F_i = \text{FFN}([\tilde{C}_i + \tilde{P}_i, \tilde{Q}]), \quad (5)$$

where  $[\cdot]$  is the concatenation operation. Taking the multi-modal interactive feature  $\{F_i\}$  as input, this layer predicts the matching score for each video segment candidate and the language query. This layer consists of two feed-forward layers followed by ReLU and Sigmoid activation respectively:

$$\hat{s}_i = \text{Sigmoid}(W_2 \text{ReLU}(W_1 F_i)), \quad (6)$$

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 ground-truth moment. 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.

### 3.3 Dynamic Adjustor

The dynamic update of our learnable proposals is performed by the dynamic adjustor. For  $N$  candidates generated before, we can obtain  $N$  scores for each candidate from the previous layer. The update strategy is that the model only adjusts the proposal with the largest score when a sample comes in, namely the most certain update. Through multiple iterations, the learned proposals can statistically represent the real distribution of the dataset. We adopt temporal IoU loss to achieve the goal:

$$\mathcal{L}_{\text{IoU}} = 1 - tIoU(\hat{b}_i, b_i), \quad (7)$$

where  $\hat{b}_i$  is the bounding box of the best-matching candidate and  $b_i$  is the ground-truth coordinates.

### 3.4 Boundary-aware Predictor

We adopt a boundary-aware predictor to further boost the performance. LPNet is still essentially a propose-and-rank approach which is hard to model the boundary information. By depicting the video as a series of segments, propose-and-rank methods break down the natural structure of videos thus causes sub-optimal results. Instead of adding additional module to explicitly incorporate boundary information, we argue that only utilizing a boundary-aware loss can significantly improve model performance. The boundary-aware predictor takes frame-level multi-modal feature as input. A bidirectional LSTM and two feed-forward layers are used to predict distribution of the start and end timestamps, *i.e.*,

$$H^s, H^e = \text{BiLSTM}(V^q), \quad (8)$$

$$L^s = \text{FFN}_s(H), L^e = \text{FFN}_e(H), \quad (9)$$

where  $H^s$  and  $H^e$  are the hidden states of the LSTM<sub>s</sub> and LSTM<sub>e</sub>, respectively.  $L^s$  and  $L^e$  denote the logits of start and end boundaries computed by two feed-forward layer. In order to avoid introducing the noise caused by label uncertainty, following (Opazo et al., 2020), we relax the ground-truth label near the start and end point and adopt KL divergence to fit distributions.

### 3.5 Training and Inference

**Training Objectives.** Each training sample consists of an untrimmed video, a language query and the corresponding ground-truth video segment. For each segment candidate, we compute the temporal IoU between the candidate and the ground-truth segment as the matching score. For each video frame with the frame-level feature, two class labels indicating whether or not the frame is the start or the end boundary are assigned. In this paper, we use soft label to avoid label uncertainty. There are two loss functions for the propose-and-rank module and boundary-aware predictor:

**Matching Regression Loss:**

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

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

**Boundary-aware Loss:**

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}(P_s || Y_s) + D_{\text{KL}}(P_e || Y_e), \quad (11)$$

where the  $D_{KL}$  is Kullback-Leibler divergence.  $Y_s$  and  $Y_e$  are ground-truth relaxed labels for the start and end boundaries, respectively.  $P_s$  and  $P_e$  are obtained from  $L^s$  and  $L^e$  via SoftMax.

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

$$\mathcal{L} = \mathcal{L}_{KL} + \lambda \times \mathcal{L}_{reg}, \quad (12)$$

where  $\lambda$  is a weight that balances the two losses. During training, we directly use the ground-truth matching score to decide which proposal to adjust by  $\mathcal{L}_{IoU}$  in Eq. (7). Specifically, the  $\mathcal{L}_{IoU}$  is used to update the parameters of proposal boxes and  $\mathcal{L}$  is used to update the rest of network.

**Inference.** Given a video, a language query and a set of learned proposal boxes, 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.

## 4 Experiments

### 4.1 Datasets

We evaluate our LPNet on two public benchmarks: 1) **Charades-STA** (Gao et al., 2017): It is built on Charades and contains 6,672 videos of daily indoor 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) **ActivityNet Captions** (Krishna et al., 2017): It contains around 20k open domain videos for video grounding task. We follow the split in (Yuan et al., 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.

### 4.2 Evaluation Metrics

Following prior works (Yuan et al., 2021), 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  $\theta$ . “mIoU” means the average IoU with ground-truth over all testing samples. In our experiments, we set  $n = 1$  and  $\theta \in \{0.3, 0.5, 0.7\}$ .

### 4.3 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 queries, we initialize each word with 300d GloVe vectors and all word embeddings are fixed during training. The dimension of the intermediate layer in LPNet is set to 256. The number of convolution blocks in embedding encoder is 4 and the kernel size is set to 7. The temporal RoIAlign length  $l$  is set to 16. To avoid elaborate design, the number of learnable proposals is uniformly set to 300 on both datasets. For all datasets, we trained the model for 100 epochs. Dropout and early stopping strategies are adopted to prevent overfitting. We implement our LPNet on Tensorflow. The  $\lambda$  in Eq. (12) is set to 100. The whole framework is trained by Adam optimizer with learning rate 0.0001.

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

**Settings.** We compare the proposed LPNet with several state-of-the-art NLVL methods on two datasets. These methods are grouped into three categories by the viewpoints of propose-and-rank and proposal-free approach: 1) propose-and-rank models: **CTRL** (Gao et al., 2017), **ROLE** (Liu et al., 2018b), **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). **2D-TAN** (Zhang et al., 2020b) 2) proposal-free models: **ABLR-af**, **ABLR-aw** (Yuan et al., 2019), **DEBUG** (Lu et al., 2019), **ExCL** (Ghosh et al., 2019), **GDP** (Chen et al., 2020), **VSLNet** (Zhang et al., 2020a), **LGI** (Mun et al., 2020), **DRN** (Zeng et al., 2020), **BPNet** (Xiao et al., 2021) 3) Others: **RWM** (He et al., 2019), **SM-RL** (Wang et al., 2019).

The results on two benchmarks are reported in Table 1 to Table 2. We can observe that our LPNet achieves new state-of-the-art performance over almost all metrics and benchmarks. Table 1 summarizes the results on Charades-STA. For a fair comparison with methods with different feature, we use both C3D and I3D (Carreira and Zisserman, 2017) video features and we report the VSLNet with C3D feature (500d by PCA) from BPNet (Xiao et al., 2021). We observe that LPNet works well in even stricter metrics, *e.g.*, LPNet achieves a significant 2.28 absolute improvement in  $IoU@0.7$  compared to the second result with I3D feature,

Methods	Feat.	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
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
LPNet	C3D	<b>59.14</b>	<b>40.94</b>	<b>21.13</b>	<b>39.67</b>
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
DRN	I3D	–	53.09	31.75	–
LPNet	I3D	<b>66.59</b>	<b>54.33</b>	<b>34.03</b>	<b>47.71</b>

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

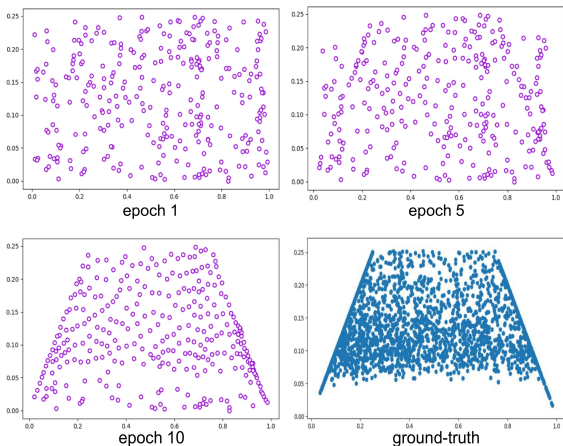


Figure 4: The distribution of learnable proposals during training process, which is getting closer to the ground-truth distribution of samples. Horizontal and vertical axes represent the normalized center coordinate and half length of proposals. We initialized the maximum length of proposals on Charades-STA as 0.5 according to its characteristics.

which demonstrates the effectiveness of our model. To be noticed, DEBUG and VSLNet utilize the backbone similar to ours adopted from QANet. DEBUG is a regression-based method and QANet is a classification-based method, which both belong to the proposal-free approach. The results show that our model not only surpass a multitude of propose-

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
DRN	–	42.49	22.25	–
LGI	58.52	41.51	23.07	41.13
BPNet	58.98	42.07	24.69	42.11
2D-TAN	59.45	44.51	<b>26.54</b>	–
LPNet	<b>64.29</b>	<b>45.92</b>	25.39	<b>44.72</b>

Table 2: Performance (%) of “R@1, IoU= $\theta$ ” and “mIoU” compared with the state-of-the-art NLVL models on ActivityNet Captions.

N	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
30	67.45	53.90	33.52	47.95
100	69.09	55.70	34.97	49.25
300	66.59	54.33	34.03	47.71

Table 3: Performance (%) of LPNet with the different number on proposals on Charades-STA.

and-rank methods by a lot, but also exceed these proposal-free methods.

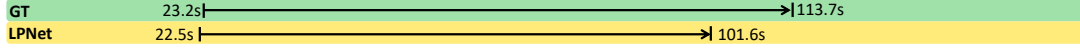
Table 2 summarizes the results with C3D features on ActivityNet Captions which has longer videos in average. Our model outperforms almost all other methods. Compared with 2D-TAN, our LPNet achieves a significant 4.48 absolute improvement in  $IoU@0.3$  but is slightly lower in  $IoU@0.7$ . This may be because the 2D-TAN enumerates much more candidates. The qualitative results of LPNet are illustrated in Figure 4. We can observe that LPNet performs well to produce the precise query-related moments.

## 5 Ablation Studies

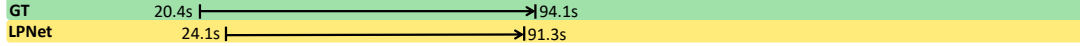
In this section, we conduct ablative experiments with different variants to better investigate our approach.

**Number of Learnable Proposals.** The number of proposals is a key factor of propose-and-rank models. We change the number of proposals of our model on Charades-STA dataset and show its impact on performance in Table 3. The results show that using only a small amount of proposals, LPNet is able to achieve impressive performance. It

**Language Query:** He picks up a sharpener and sharpens the blade of the knife on the sharpener.



**Language Query:** A wooden contraption signals that a fish has been caught, The man pulls the fish out of the water.



**Language Query:** People rides bumper cars in the carnival, a yellow car cross two girls in a grey car.

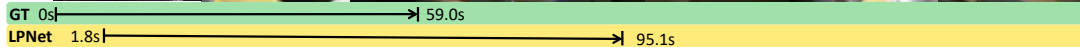


Figure 5: The qualitative results of LPNet on ActivityNet Caption dataset.

MHSA	BAL	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
		65.38	51.05	29.33	45.21
✓		66.48	52.42	30.99	45.87
	✓	66.48	54.03	34.22	47.45
✓	✓	66.59	54.33	34.03	47.71

Table 4: Performance (%) comparisons on CharadesSTA in ablative experiments of component of LPNet. MHSA: multi-head self-attention over proposal features, BAL: boundary-aware loss.

should be noted that we simply place 300 learnable proposals on both datasets in Table 1 and Table 2 to avoid artificial design. However, a smaller amount (100) of proposals get better result on CharadesSTA.

**With vs Without Boundary-aware Loss.** From Table 4, we find that there is huge improvement when the boundary-aware loss is applied. The main reason is that the KL-divergence loss utilizes frame-level information to regularize the training of the model and force the model to consider the video as a whole.

**With vs Without Multi-head Self-attention.** Comparing the first two rows in Table 4, we observe that applying multi-head self-attention mechanism to proposal features can improve the performance. This operation successfully learns the latent relations between the proposals which is helpful

to the localization task. However, when boundary-aware loss has been already applied (last two rows in Table 4), the results are very close. This may indicate that the boundary-aware loss makes the similar kind of contribution to the model.

## 6 Conclusions

In this paper, we present a novel propose-and-rank model with learnable moment proposals for NLVL. Compared to the existing propose-and-rank method with predefined temporal boxes, our model improves the performance significantly because 1) our model disengages from the hand-designed rules for bounding boxes so that it can produce more accurate temporal boundaries; 2) sparse sampled candidate release the pressure for subsequent ranking process; 3) boundary-aware loss regularize the model to avoid sub-optimum. In the future, we are going to explore a more effective way to learn better proposals and extend this idea to other tasks.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (U19B2043, 61976185), Zhejiang Natural Science Foundation (LR19F020002), Zhejiang Innovation Foundation(2019R52002), and the Fundamental Research Funds for the Central Universities.



## References

- Meng Cao, Long Chen, Mike Zheng Shou, Can Zhang, and Yuexian Zou. 2021. On pursuit of designing multi-modal transformer for video grounding. In *EMNLP*.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *ECCV*, pages 213–229.
- João Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733.
- Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. 2018. Temporally grounding natural sentence in video. In *EMNLP*, pages 162–171.
- Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chile Tan, and Xiaolin Li. 2020. Rethinking the bottom-up framework for query-based video localization. In *AAAI*, pages 10551–10558.
- Shaoxiang Chen and Yu-Gang Jiang. 2019. Semantic proposal for activity localization in videos via sentence query. In *AAAI*, pages 8199–8206.
- Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.
- Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. 2019. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6568–6577.
- Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. TALL: temporal activity localization via language query. In *ICCV*, pages 5277–5285.
- Kaifeng Gao, Long Chen, Yifeng Huang, and Jun Xiao. 2021. Video relation detection via tracklet based visual transformer. In *ACM MM*.
- Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. 2019. MAC: mining activity concepts for language-based temporal localization. In *WACV*, pages 245–253.
- Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander G. Hauptmann. 2019. Excl: Extractive clip localization using natural language descriptions. In *NAACL*, pages 1984–1990.
- Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. 2019. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *AAAI*, pages 334–343.
- Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. 2017. Localizing moments in video with natural language. In *ICCV*, pages 5804–5813.
- Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. 2018. Localizing moments in video with temporal language. In *EMNLP*, pages 1380–1390.
- Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*, pages 706–715.
- Hei Law and Jia Deng. 2018. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781.
- Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2018. TVQA: localized, compositional video question answering. In *EMNLP*, pages 1369–1379.
- Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. 2018a. Attentive moment retrieval in videos. In *SIGIR*, pages 15–24.
- Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. 2018b. Cross-modal moment localization in videos. In *MM*, pages 843–851.
- Chujie Lu, Long Chen, Chile Tan, Xiaolin Li, and Jun Xiao. 2019. DEBUG: A dense bottom-up grounding approach for natural language video localization. In *EMNLP*, pages 5143–5152.
- Jonghwan Mun, Minsu Cho, and Bohyung Han. 2020. Local-global video-text interactions for temporal grounding. In *CVPR*, pages 10807–10816.
- Cristian Rodriguez Opazo, Edison Marrese-Taylor, Fatemeh Sadat Saleh, Hongdong Li, and Stephen Gould. 2020. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *WACV*, pages 2453–2462.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.
- Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99.
- Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. 2018. Find and focus: Retrieve and localize video events with natural language queries. In *ECCV*, pages 202–218.

- Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. 2021. Sparse R-CNN: end-to-end object detection with learnable proposals. In *CVPR*, pages 14454–14463.
- Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497.
- Weining Wang, Yan Huang, and Liang Wang. 2019. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *CVPR*, pages 334–343.
- Wenxiao Wang, Lu Yao, Long Chen, Deng Cai, Xiaofei He, and Wei Liu. 2021. Crossformer: A versatile vision transformer based on cross-scale attention. *arXiv preprint arXiv:2108.00154*.
- Shaoning Xiao, Long Chen, Songyang Zhang, Wei Ji, Jian Shao, Lu Ye, and Jun Xiao. 2021. Boundary proposal network for two-stage natural language video localization. In *AAAI*, pages 2986–2994.
- Huijuan Xu, Kun He, Bryan A. Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. 2019. Multi-level language and vision integration for text-to-clip retrieval. In *AAAI*, pages 9062–9069.
- Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. 2017. Video question answering via attribute-augmented attention network learning. In *SIGIR*, pages 829–832.
- Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.
- Yitian Yuan, Xiaohan Lan, Long Chen, Wei Liu, Xin Wang, and Wenwu Zhu. 2021. A closer look at temporal sentence grounding in videos: Datasets and metrics. *arXiv preprint arXiv:2101.09028*.
- Yitian Yuan, Tao Mei, and Wenwu Zhu. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*, pages 9159–9166.
- Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. 2020. Dense regression network for video grounding. In *CVPR*, pages 10284–10293.
- Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S. Davis. 2019. MAN: moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*, pages 1247–1257.
- Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. 2020a. Span-based localizing network for natural language video localization. In *ACL*, pages 6543–6554.
- Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. 2020b. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, pages 12870–12877.