Adaptive Proposal Generation Network for Temporal Sentence Localization in Videos

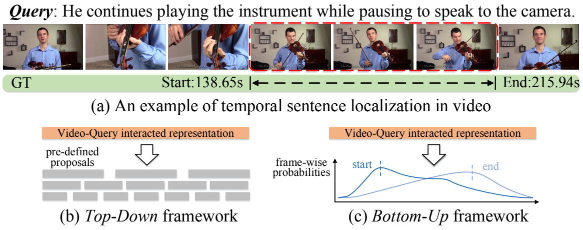
Daizong Liu[1,2*], Xiaoye Qu[3*], Jianfeng Dong[4], Pan Zhou[1†]
[1]The Hubei Engineering Research Center on Big Data Security, School of
Cyber Science and Engineering, Huazhong University of Science and Technology
[2]School of Electronic Information and Communication,
Huazhong University of Science and Technology
[3]Huawei Cloud [4]Zhejiang Gongshang University
{dzliu,panzhou}@hust.edu.cn, quxiaoye@huawei.com, dongjf24@gmail.com

Abstract

We address the problem of temporal sentence localization in videos (TSLV). Traditional methods follow a *top-down* framework which localizes the target segment with pre-defined segment proposals. Although they have achieved decent performance, the proposals are handcrafted and redundant. Recently, *bottom-up* framework attracts increasing attention due to its superior efficiency. It directly predicts the probabilities for each frame as a boundary. However, the performance of *bottom-up* model is inferior to the *top-down* counterpart as it fails to exploit the segment-level interaction. In this paper, we propose an **Adaptive Proposal Generation Network** (APGN) to maintain the segment-level interaction while speeding up the efficiency. Specifically, we first perform a foreground-background classification upon the video and regress on the foreground frames to adaptively generate proposals. In this way, the handcrafted proposal design is discarded and the redundant proposals are decreased. Then, a proposal consolidation module is further developed to enhance the semantic of the generated proposals. Finally, we locate the target moments with these generated proposals following the *top-down* framework. Extensive experiments on three challenging benchmarks show that our proposed APGN significantly outperforms previous state-of-the-art methods.

1 Introduction

Temporal sentence localization in videos is an important yet challenging task in natural language processing, which has drawn increasing attention over the last few years due to its vast potential applications in information retrieval (Dong et al., 2019; Yang et al., 2020) and human-computer interaction (Singha et al., 2018). It aims to ground the most relevant video segment according to a given sentence query. As shown in Figure 1 (a), most parts of video contents are irrelevant to the query (background) while only a short segment matches it (foreground). Therefore, video and query information need to be deeply incorporated to distinguish the fine-grained details of different video segments.


Query: He continues playing the instrument while pausing to speak to the camera.
GT
Start:138.65s
End:215.94s
Video-Query interacted representation
pre-defined proposals
Video-Query interacted representation
frame-wise probabilities
start
end
(a) An example of temporal sentence localization in video

(b) *Top-Down* framework

(c) *Bottom-Up* framework

Figure 1: (a) An example of temporal sentence localization in videos. (b) The *Top-Down* framework predicts the confidence scores of a large number of pre-defined proposals for ranking. (c) The *Bottom-Up* framework regresses the probabilities of all frames as start or end boundaries.

Most previous works (Gao et al., 2017; Chen et al., 2018; Zhang et al., 2019; Yuan et al., 2019a; Zhang et al., 2020b; Liu et al., 2021, 2020a,b) follow the *top-down* framework which pre-defines a large set of segment candidates (a.k.a proposals) in the video with sliding windows, and measures the similarity between the query and each candidate. The best segment is then selected according to the similarity. Although these methods achieve significant performance, they are sensitive to the proposal quality and present slow localization speed due to redundant proposals. Recently, several works (Rodriguez et al., 2020; Zhang et al., 2020a; Yuan et al., 2019b) exploit the *bottom-up* framework which directly predicts the probabilities of each frame as the start or end boundaries of segment. These methods are proposal-free and much more efficient. However, they neglect the rich information between start and end boundaries without capturing the segment-level interaction. Thus, the performance of *bottom-up* models is behind the performance of *top-down* counterpart thus far.

[*]Equal contributions. [†]Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9292–9301
November 7–11, 2021.

To avoid the inherent drawbacks of proposal design in the *top-down* framework and maintain the localization performance, in this paper, we propose an adaptive proposal generation network (APGN) for an efficient and effective localization approach. Firstly, we perform boundary regression on the foreground frames to generate proposals, where foreground frames are obtained by a foreground-background classification on the entire video. In this way, the noisy responses on the background frames are attenuated, and the generated proposals are more adaptive and discriminative compared to the pre-defined ones. Secondly, we perform proposal ranking to select target segment in a *top-down* manner upon these generative proposals. As the number of proposals is much fewer than the pre-defined methods, the ranking stage is more efficient. Furthermore, we additionally consider the proposal-wise relations to distinguish their fine-grained semantic details before the proposal ranking stage.

To achieve the above framework, APGN first generates query-guided video representations after encoding video and query features and then predicts the foreground frames using a binary classification module. Subsequently, a regression module is utilized to generate a proposal on each foreground frame by regressing the distances from itself to start and end segment boundaries. After that, each generated proposal contains independent coarse semantic. To capture higher-level interactions among proposals, we encode proposal-wise features by incorporating both positional and semantic information, and represent these proposals as nodes to construct a proposal graph for reasoning correlations among them. Consequently, each updated proposal obtains more fine-grained details for following boundary refinement process.

Our contributions are summarized as follows:

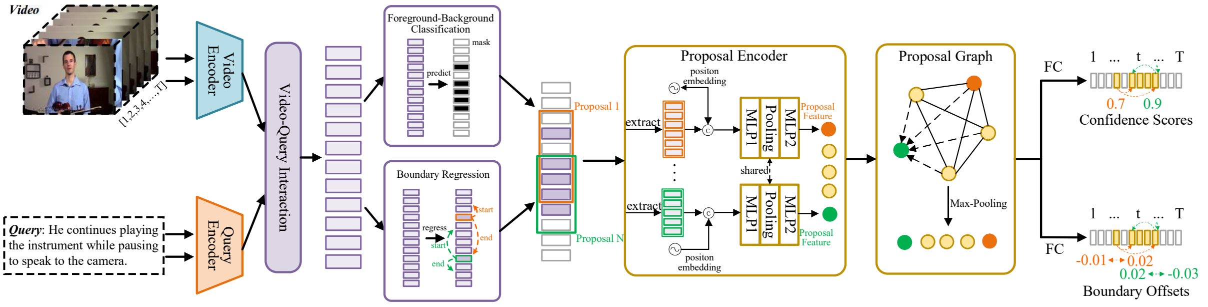
- We propose an adaptive proposal generation network (APGN) for TSLV task, which adaptively generates discriminative proposals without handcrafted design, thus making localization both effective and efficient.
- To further refine the semantics of the generated proposals, we introduce a proposal graph to consolidate proposal-wise features by reasoning their higher-order relations.
- We conduct experiments on three challenging datasets (ActivityNet Captions, TACoS, and Charades-STA), and results show that our proposed APGN significantly outperforms the existing state-of-the-art methods.

2 Related Work

Temporal sentence localization in videos is a new task introduced recently (Gao et al., 2017; Anne Hendricks et al., 2017), which aims to localize the most relevant video segment from a video with sentence descriptions. Various algorithms (Anne Hendricks et al., 2017; Gao et al., 2017; Chen et al., 2018; Zhang et al., 2019; Yuan et al., 2019a; Zhang et al., 2020b; Qu et al., 2020; Yang et al., 2021) have been proposed within the *top-down* framework, which samples candidate segments from a video first, then integrates the sentence representation with those video segments individually and evaluates their matching relationships. Some of them (Anne Hendricks et al., 2017; Gao et al., 2017) propose to use the sliding windows as proposals and then perform a comparison between each proposal and the input query in a joint multi-modal embedding space. To improve the quality of the proposals, (Zhang et al., 2019; Yuan et al., 2019a) pre-cut the video on each frame by multiple pre-defined temporal scale, and directly integrate sentence information with fine-grained video clip for scoring. (Zhang et al., 2020b) further build a 2D temporal map to construct all possible segment candidates by treating each frame as the start or end boundary, and match their semantics with the query information. Although these methods achieve great performance, they are severely limited by the heavy computation on proposal matching/ranking, and sensitive to the quality of pre-defined proposals.

Recently, many methods (Rodriguez et al., 2020; Chen et al., 2020; Yuan et al., 2019b; Mun et al., 2020; Zeng et al., 2020; Zhang et al., 2020a; Nan et al., 2021) propose to utilize the *bottom-up* framework to overcome above drawbacks. They do not rely on the segment proposals and directly select the starting and ending frames by leveraging cross-modal interactions between video and query. Specifically, they predict two probabilities at each frame, which indicate whether this frame is a start or end frame of the ground truth video segment. Although these methods perform segment localization more efficiently, they lose the segment-level interaction, and the redundant regression on background frames may provide disturbing noise for boundary decision, leading to worse localization performance than *top-down* methods.

In this paper, we propose to preserve the segment-level interaction while speeding up the



(a) Query-guided video feature generation (b) Proposal generation (c) Proposal consolidation (d) Localization

Figure 2: Overall architecture of APGN. (a) Given a video and a query, we first encode and interact them to obtain query-guided video features. (b) Then, along with regressing boundaries on each frame, we perform foreground-background classification to identify the foreground frames whose corresponding predicted boundaries are further taken as the generated segment proposals. (c) We further encode each proposal and refine them using a graph convolutional network. (d) At last, we predict the confidence score and boundary offset for each proposal.

localization efficiency. Specifically, we design a binary classification module on the entire video to filter out the background responses, which helps model focus more on the discriminative frames. At the same time, we replace the pre-defined proposals with the generated ones and utilize a proposal graph for refinement.

3 The Proposed Method

3.1 Overview

Given an untrimmed video V and a sentence query Q, the TSLV task aims to localize the start and end timestamps (τ_s, τ_e) of a specific video segment referring to the sentence query. We focus on addressing this task by adaptively generating proposals. To this end, we propose a binary classification module to filter out the redundant responses on background frames. Then, each foreground frame with its regressed start-end boundaries are taken as the generated segment proposal. In this way, the number of the generated proposals is much smaller than the number of pre-defined ones, making the model more efficient. Besides, a proposal graph is further developed to refine proposal features by learning their higher-level interactions. Finally, the confidence score and boundary offset are predicted for each proposal. Figure 2 illustrates the overall architecture of our APGN.

3.2 Feature Encoders

Video encoder. Given a video V, we represent it as $V = \{v_t\}_{t=1}^{T}$, where v_t is the t-th frame and T is the length of the entire video. We first extract the features by a pre-trained network, and then employ a self-attention (Vaswani et al., 2017) module to capture the long-range dependencies among video frames. We also utilize a Bi-GRU (Chung et al., 2014) to learn the sequential characteristic. The final video features are denoted as $\boldsymbol{V} = \{\boldsymbol{v}_t\}_{t=1}^{T} \in \mathbb{R}^{T\times D}$, where D is the feature dimension.

Query encoder. Given a query $Q = \{q_n\}_{n=1}^{N}$, where q_n is the n-th word and N is the length of the query. Following previous works (Zhang et al., 2019; Zeng et al., 2020), we first generate the word-level embeddings using Glove (Pennington et al., 2014), and also employ a self-attention module and a Bi-GRU layer to further encode the query features as $\boldsymbol{Q} = \{\boldsymbol{q}_n\}_{n=1}^{N} \in \mathbb{R}^{N\times D}$.

Video-Query interaction. After obtaining the encoded features $\boldsymbol{V}, \boldsymbol{Q}$, we utilize a co-attention mechanism (Lu et al., 2019) to capture the cross-modal interactions between video and query features. Specifically, we first calculate the similarity scores between $\boldsymbol{V}$ and $\boldsymbol{Q}$ as:

$$\boldsymbol{S} = \boldsymbol{V}(\boldsymbol{Q}\boldsymbol{W}_S)^{\mathrm{T}} \in \mathbb{R}^{T\times N}, \tag{1}$$

where $\boldsymbol{W}_S \in \mathbb{R}^{D\times D}$ projects the query features into the same latent space as the video. Then, we compute two attention weights as:

$$\boldsymbol{A} = \boldsymbol{S}_r(\boldsymbol{Q}\boldsymbol{W}_S) \in \mathbb{R}^{T\times D}, \boldsymbol{B} = \boldsymbol{S}_r\boldsymbol{S}_c^{\mathrm{T}}\boldsymbol{V} \in \mathbb{R}^{T\times D}, \tag{2}$$

where $\boldsymbol{S}_r$ and $\boldsymbol{S}_c$ are the row- and column-wise softmax results of $\boldsymbol{S}$, respectively. We compose the final query-guided video representation by learning its sequential features as follows:

$$\widetilde{\boldsymbol{V}} = \mathrm{BiGRU}([\boldsymbol{V}; \boldsymbol{A}; \boldsymbol{V}\odot\boldsymbol{A}; \boldsymbol{V}\odot\boldsymbol{B}]) \in \mathbb{R}^{T\times D}, \tag{3}$$

where $\widetilde{\boldsymbol{V}} = \{\widetilde{\boldsymbol{v}}_t\}_{t=1}^{T}$, BiGRU$(\cdot)$ denotes the Bi-GRU layers, $[;]$ is the concatenate operation, and $\odot$ is the element-wise multiplication.

3.3 Proposal Generation

Given the query-guided video features $\widetilde{\boldsymbol{V}}$, we aim to generate the proposal tuple (t, l_s^t, l_e^t) based on each foreground frame v_t, where l_s^t, l_e^t denotes the distances from frame v_t to the starting and ending segment boundaries, respectively. To this end, we first perform binary classification on the whole frames to distinguish the foreground and background frames, and then treat the foreground ones as positive samples and regress the segment boundaries on these frames as generated proposals.

Foreground-Background classification. In the TSLV task, most videos are more than two minutes long while the lengths of annotated target segments only range from several seconds to one minute (e.g. on ActivityNet Caption dataset). Therefore, there exists much noises from the background frames which may disturb the accurate segment localization. To alleviate it, we first classify the background frames and filter out their responses in latter regression. By distinguishing the foreground and background frames with annotations, we design a binary classification module with three full-connected (FC) layers to predict the class y_t on each video frame. Considering the unbalanced foreground/background distribution, we formulate the balanced binary cross-entropy loss as:

$$\mathcal{L}_{class} = -\sum_{t=1}^{T_{back}} \frac{T_{back}}{T}\log(y_t) - \sum_{t=1}^{T_{fore}} \frac{T_{fore}}{T}\log(1-y_t), \tag{4}$$

where T_{fore}, T_{back} are the numbers of foreground and background frames. T is the number of total video frames. Therefore, we can differentiate between frames from foreground and background during both training and testing.

Boundary regression. With the query-guided video representation $\widetilde{\boldsymbol{V}}$ and the predicted binary sequence of 0-1, we then design a boundary regression module to predict the distance from each foreground frame to the start (or end) frame of the video segment that corresponds to the query. We implement this module by three 1D convolution layers with two output channels. Given the predicted distance pair (l_s^t, l_e^t) and ground-truth distance (g_s^t, g_e^t), we define the regression loss as:

$$\mathcal{L}_{reg} = \frac{1}{T_{fore}} \sum_{t=1}^{T_{fore}} (1 - \text{IoU}((t, l_s^t, l_e^t), (t, g_s^t, g_e^t))), \tag{5}$$

where $\text{IoU}(\cdot)$ computes the Intersection over Union (IoU) score between the predicted segment and its ground-truth. After that, we can represent the generated proposal as tuples $\{(t, l_s^t, l_e^t)\}_{t=1}^{T_{fore}}$ based on the regression results of the foreground frames.

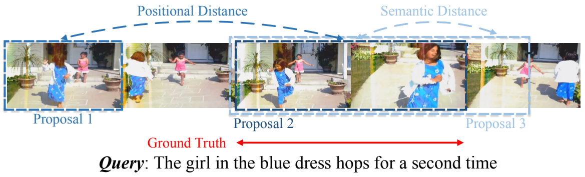


Figure 3: To distinguish above three proposals, both positional and semantic relations among proposals needs to be considered.

3.4 Proposal Consolidation

So far, we have generated a certain number of proposals that are significantly less than the predefined ones in existing *top-down* framework, making the final scoring and ranking process much efficient. To further refine the proposal features for more accurate segment localization, we explicitly model higher-order interactions between the generated proposals to learn their relations. As shown in Figure 3, proposal 1 and proposal 2 contain same semantics of “blue" and “hops", we need to model their positional distance to distinguish them and refine their features for better understanding the phrase “second time". Also, for the proposals (proposal 2 and 3) which are local neighbors, we have to learn their semantic distance to refine their representations. Therefore, in our APGN, we first encode each proposal feature with both positional embedding and frame-wise semantic features, and then define a graph convolutional network (GCN) over the proposals for proposal refinement.

Proposal encoder. For each proposal tuple (t, l_s^t, l_e^t), we represent its segment boundary as $(t - l_s^t, t + l_e^t)$. Before aggregating the features of its contained frames within this segment boundary, we first concatenate a position embedding $\boldsymbol{emb}_t^{pos}$ to each frame-wise feature $\widetilde{\boldsymbol{v}}_t$, in order to inject position information on frame t as follows:

$$\widetilde{\boldsymbol{v}}'_t = [\widetilde{\boldsymbol{v}}_t; \boldsymbol{emb}_t^{pos}] \in \mathbb{R}^{1\times(D+d)}, \tag{6}$$

where $\boldsymbol{emb}_t^{pos}$ denotes the position embedding of the t-th position, and d is the dimension of $\boldsymbol{emb}_t^{pos}$. We follow (Vaswani et al., 2017) and use the sine and cosine functions of different frequencies to compose position embeddings:

$$\boldsymbol{emb}_t^{pos}[2j] = \sin(\frac{t}{10000^{2j/d}}), \tag{7}$$

$$\boldsymbol{emb}_t^{pos}[2j+1] = \cos(\frac{t}{10000^{2j/d}}), \tag{8}$$

where $2j$ and $2j+1$ are the even and odd indices of the position embedding. In this way, each dimension of the positional encoding corresponds to a sinusoid, allowing the model to easily learn to attend to absolute positions. Given the frame features $\{\widetilde{\boldsymbol{v}}'_t\}_{t=1}^{T_{fore}}$ and a proposal segment $(t - l_s^t, t + l_e^t)$, we encode the vector feature $\boldsymbol{p}_t$ of t-th proposal by aggregating the features of the contained frames in the segment as:

$$\boldsymbol{p}_t = \text{MLP}_2(\text{Pool}(\text{MLP}_1([\widetilde{\boldsymbol{v}}'_{\lceil t-l_s^t \rceil}, ..., \widetilde{\boldsymbol{v}}'_{\lfloor t+l_e^t \rfloor}]))), \tag{9}$$

where each MLP has two FC layers, Pool$(\cdot)$ denotes the max-pooling. The frames from each proposal are independently processed by MLP$_1$ before being pooled (channel-wise) to a single feature vector and passed to MLP$_2$ where information from different frames are further combined. Thus, we can represent the encoded proposal feature as $\boldsymbol{p}_t \in \mathbb{R}^{1\times(D+d)}$.

Proposal graph. We construct a graph over the proposal features $\{\boldsymbol{p}_t\}_{t=1}^{T_{fore}}$, where each node of the graph is a proposal associated with both positions and semantic features. We full connect all node pairs, and define relations between each proposal-pair $(\boldsymbol{p}_t, \boldsymbol{p}_{t'})$ for edge convolution (Wang et al., 2018) as:

$$\boldsymbol{e}_{t,t'} = \text{Relu}(\boldsymbol{p}_t \boldsymbol{\theta}_1 + (\boldsymbol{p}_{t'} - \boldsymbol{p}_t)\boldsymbol{\theta}_2), \tag{10}$$

where $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ are learnable parameters. We update each proposal feature $\boldsymbol{p}_t$ to $\widehat{\boldsymbol{p}}_t$ as follow:

$$\widehat{\boldsymbol{p}}_t = \text{MaxPool}(\boldsymbol{e}_t), \;\; \boldsymbol{e}_t = \{\boldsymbol{e}_{t,t'}\}_{t'=1}^{T_{fore}}. \tag{11}$$

This GCN module consists of k stacked graph convolutional layers. After the above proposal consolidation with graph, we are able to learn the refined proposal features.

3.5 Localization Head

After proposal consolidation, we feed the refined features $\widehat{\boldsymbol{P}} = \{\widehat{\boldsymbol{p}}_t\}_{t=1}^{T_{fore}}$ into two separate heads to predict their confidence scores and boundary offsets for proposal ranking and refinement. Specifically, we employ two MLPs on each feature $\widehat{\boldsymbol{p}}_t$ as:

$$r_t = \text{Sigmoid}(\text{MLP}_3(\widehat{\boldsymbol{p}}_t)), \tag{12}$$

$$(\delta_s^t, \delta_e^t) = \text{MLP}_4(\widehat{\boldsymbol{p}}_t), \tag{13}$$

where $r_t \in (0, 1)$ is the confidence score, and (δ_s^t, δ_e^t) is the offsets. Therefore, the final predicted segment of proposal t can be represented as $(t - l_s^t + \delta_s^t, t + l_e^t + \delta_e^t)$. To learn the confidence scoring rule, we first compute the IoU score o_t between each proposal segment with the ground-truth (τ_s, τ_e), then we adopt the alignment loss function as below:

$$\mathcal{L}_{align} = -\frac{1}{T_{fore}} \sum_{t=1}^{T_{fore}} o_t \log(r_t) + (1-o_t)\log(1-r_t). \tag{14}$$

Given the ground-truth boundary offsets $(\hat{\delta}_s^t, \hat{\delta}_e^t)$ of proposal t, we also fine-tune its offsets by a boundary loss as:

$$\mathcal{L}_b = \frac{1}{T_{fore}} \sum_{t=1}^{T_{fore}} \text{SL}_1(\hat{\delta}_s^t - \delta_s^t) + \text{SL}_1(\hat{\delta}_e^t - \delta_e^t), \tag{15}$$

where SL$_1(\cdot)$ denotes the smooth L1 loss function.

At last, our APGN model is trained end-to-end from scratch using the multi-task loss :

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{class} + \lambda_2 \cdot \mathcal{L}_{reg} + \lambda_3 \cdot \mathcal{L}_{align} + \lambda_4 \cdot \mathcal{L}_b. \tag{16}$$

4 Experiments

4.1 Datasets and Evaluation

ActivityNet Captions. It is a large dataset (Krishna et al., 2017) which contains 20k videos with 100k language descriptions. This dataset pays attention to more complicated human activities in daily life. Following public split, we use 37,417, 17,505, and 17,031 sentence-video pairs for training, validation, and testing, respectively.

TACoS. This dataset (Regneri et al., 2013) collects 127 long videos, which are mainly about cooking scenarios, thus lacking the diversity. We use the same split as (Gao et al., 2017), which has 10146, 4589 and 4083 sentence-video pairs for training, validation, and testing, respectively.

Charades-STA. (Gao et al., 2017) consists of 9,848 videos of daily life indoors activities. There are 12,408 sentence-video pairs for training and 3,720 pairs for testing.

Evaluation Metric. Following (Zhang et al., 2019; Zeng et al., 2020), we adopt "R@n, IoU=m" as our evaluation metrics, which is defined as the percentage of at least one of top-n selected moments having IoU larger than m.

4.2 Implementation Details

Following (Zhang et al., 2020b; Zeng et al., 2020), for video input, we apply a pre-trained C3D network for all three datasets to obtain embedded features. We also extract the I3D (Carreira and Zisserman, 2017) and VGG (Simonyan and Zisserman,

Method	Feature	R@1, IoU=0.5	R@1, IoU=0.7	R@5, IoU=0.5	R@5 IoU=0.7
TGN	C3D	28.47	-	43.33	-
CTRL	C3D	29.01	10.34	59.17	37.54
QSPN	C3D	33.26	13.43	62.39	40.78
CBP	C3D	35.76	17.80	65.89	46.20
SCDM	C3D	36.75	19.86	64.99	41.53
GDP	C3D	39.27	-	-	-
LGI	C3D	41.51	23.07	-	-
VSLNet	C3D	43.22	26.16	-	-
CMIN	C3D	43.40	23.88	67.95	50.73
DRN	C3D	45.45	24.36	77.97	50.30
2DTAN	C3D	44.51	26.54	77.13	61.96
APGN	C3D	**48.92**	**28.64**	**78.87**	**63.19**

Table 1: Performance compared with the state-of-the-art TSLV models on ActivityNet Captions dataset.

2014) features on Charades-STA. After that, we apply PCA to reduce their feature dimension to 500 for decreasing the model parameters. We set the length of video to 200 for ActivityNet Caption and TACoS, 64 for Charades-STA. For sentence input, we utilize Glove model to embed each word to 300 dimension features. The dimension D is set to 512, d is set to 256. The number of graph layer is $k = 2$. We set the batchsize as 64. We train our model with an Adam optimizer for 100 epochs. The initial learning rate is set to 0.0001 and it is divided by 10 when the loss arrives on plateaus. $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ in the loss function are 0.1, 1, 1, 1 and decided by the weight magnitude.

4.3 Performance Comparison

Compared methods. We compare our proposed APGN with state-of-the-art methods. We group them into: (1) *top-down* methods: TGN (Chen et al., 2018), CTRL (Gao et al., 2017), QSPN (Xu et al., 2019), CBP (Wang et al., 2020), SCDM (Yuan et al., 2019a), CMIN (Zhang et al., 2019), and 2DTAN (Zhang et al., 2020b). (2) *bottom-up* methods: GDP (Chen et al., 2020), LGI (Mun et al., 2020), VSLNet (Zhang et al., 2020a), DRN (Zeng et al., 2020).

Quantitative comparison. As shown in Table 1, 2 and 3, our APGN outperforms all the existing methods by a large margin. Specifically, on ActivityNet Caption dataset, compared to the previous best *top-down* method 2DTAN, we do not rely on large numbers of pre-defined and outperform it by 4.41%, 2.10%, 1.74%, 1.23% in all metrics, respectively. Compared to the previous best *bottom-up* method DRN, our APGN brings significant improvement of 4.28% and 12.89% in the strict “R@1, IoU=0.7” and “R@5, IoU=0.7” metrics, respectively. Although TACoS suffers from similar kitchen background and cooking objects among the videos, it is worth noting that our APGN still achieves significant improvements. On Charades-STA dataset, for fair comparisons with other methods, we perform experiments with same features (i.e., VGG, C3D, and I3D) reported in their papers. It shows that our APGN reaches the highest results over all evaluation metrics.

Method	Feature	R@1, IoU=0.3	R@1, IoU=0.5	R@5, IoU=0.3	R@5, IoU=0.5
TGN	C3D	21.77	18.90	39.06	31.02
CTRL	C3D	18.32	13.30	36.69	25.42
QSPN	C3D	20.15	15.23	36.72	25.30
CBP	C3D	27.31	24.79	43.64	37.40
SCDM	C3D	26.11	21.17	40.16	32.18
GDP	C3D	24.14	-	-	-
VSLNet	C3D	29.61	24.27	-	-
CMIN	C3D	24.64	18.05	38.46	27.02
DRN	C3D	-	23.17	-	33.36
2DTAN	C3D	37.29	25.32	57.81	45.04
APGN	C3D	**40.47**	**27.86**	**59.98**	**47.12**

Table 2: Performance compared with the state-of-the-art TSLV models on TACoS datasets.

Method	Feature	R@1, IoU=0.5	R@1, IoU=0.7	R@5, IoU=0.5	R@5, IoU=0.7
2DTAN	VGG	39.81	23.25	79.33	51.15
APGN	VGG	**44.23**	**25.64**	**89.51**	**57.87**
CTRL	C3D	23.63	8.89	58.92	29.57
QSPN	C3D	35.60	15.80	79.40	45.40
CBP	C3D	36.80	18.87	70.94	50.19
GDP	C3D	39.47	18.49	-	-
APGN	C3D	**48.20**	**29.37**	**89.05**	**58.49**
DRN	I3D	53.09	31.75	89.06	60.05
SCDM	I3D	54.44	33.43	74.43	58.08
LGI	I3D	59.46	35.48	-	-
APGN	I3D	**62.58**	**38.86**	**91.24**	**62.11**

Table 3: Performance compared with the state-of-the-art TSLV models on Charades-STA datasets.

Comparison on efficiency. We compare the efficiency of our APGN with previous methods on a single Nvidia TITAN XP GPU on the TACoS dataset. As shown in Table 4, it can be observed that we achieve much faster processing speeds and relatively less learnable parameters. The reason mainly owes to two folds: First, APGN generates proposals without processing overlapped sliding windows as CTRL, and generates less proposals than pre-defined methods such as 2DTAN and CMIN, thus is more efficient; Second, APGN does not apply many convolution layers like 2DTAN or multi-level feature fusion modules as DRN for cross-modal interaction, thus has less parameters.

	ACRN	CTRL	TGN	2DTAN	CMIN	DRN	**APGN**
VPS ↑	0.23	0.45	1.09	1.75	81.29	133.38	**146.67**
Para. ↓	128	**22**	166	363	78	214	91

Table 4: Efficiency comparison in terms of video per second (VPS) and parameters (Para.), where our method APGN is much efficient.

Model	class.	reg.	p.e.	graph	R@1, IoU=0.5	R@1, IoU=0.7
①	×	×	×	×	39.16	19.68
②	✓	×	×	×	40.84	21.30
③	✓	✓	×	×	42.77	23.52
④	✓	✓	✓	×	43.95	24.66
⑤	✓	✓	×	✓	45.81	26.34
⑥	✓	✓	✓	✓	**48.92**	**28.64**

Table 5: Main ablation studies on ActivityNet Caption dataset, where 'class.' and 'reg.' denotes the classification and regression modules (Sec. 3.3), 'p.e' denotes the proposal encoder (Sec. 3.4), 'graph' denotes the proposal graph (Sec. 3.4).

4.4 Ablation Study

Main ablation. As shown in Table 5, we verify the contribution of each part in our model. Starting from the backbone model (Figure 2 (a)), we first implement the baseline model ① by directly adding the *top-down* localization head ((Figure 2 (d))). In this model, we adopt pre-defined proposals as (Zhang et al., 2019). After adding the binary classification module in ②, we can find that classification module effectively filters out redundant pre-defined proposals on large number of background frames. When further applying adaptive proposal generation as ③, the generated proposals perform better than the pre-defined one ②. Note that, in ③, we directly encode proposal-wise features by max-pooling, and the classification module also makes the contribution for filtering out the negative generated proposals. To capture more fine-grained semantics for proposal refinement, we introduce a proposal encoder (model ④) for discriminative feature aggregation and a proposal graph (model ⑤) for proposal-wise feature interaction. Although each of them can only bring about 1-3% improvement, the performance increases significantly when utilizing both of them (model ⑥).

Investigation on the video/query encoder. To investigate whether a Transformer (Vaswani et al., 2017) can boost our APGN, we replace the GRU in video/query encoder with a simple Transformer and find some improvements. However, it brings larger model parameters and lower speed.

Components	VPS ↑	Para. ↓	R@1, IoU=0.5	R@1, IoU=0.7
w/. GRU	**146.67**	**91**	48.92	28.64
w/. Transformer	129.38	138	**50.11**	**29.43**

Table 6: Investigation on video and query encoders on ActivityNet Caption dataset.

Components	Module	R@1, IoU=0.5	R@1, IoU=0.7
binary classification	w/o balanced loss	46.88	27.13
	w/ balanced loss	**48.92**	**28.64**

Table 7: Investigation on binary classification on ActivityNet Caption dataset.

Components	Module	R@1, IoU=0.5	R@1, IoU=0.7
proposal encoder	w/o position	46.46	26.69
	w/ position	**48.92**	**28.64**
	w/ mean pooling	47.41	27.86
	w/ max pooling	**48.92**	**28.64**

Table 8: Investigation on proposal encoder on ActivityNet Caption dataset.

Effect of unbalanced loss. In the binary classification module, we formulate the typical loss function into a balanced one. As shown in Table 7, the model w/ balanced loss has great improvement (2.04%, 1.51%) compared to the w/o variant, which demonstrates that it is important to consider the unbalanced distribution in the classification process.

Investigation on proposal encoder. In proposal encoder, we discard the positional embedding as w/o position, and also replace the max-pooling with the mean-pooling as w/ mean pooling. From the Table 8, we can observe that positional embedding helps to learn the temporal distance (boost 2.46%, 1.95%), and the max-pooling can aggregate more discriminative features (boost 1.49%, 0.78%) than the mean-pooling.

Investigation on proposal graph. In the table 9, we also give the analysis on the proposal graph. Compared to w/ edge convolution model (Wang et al., 2018), w/ edge attention directly utilizes co-attention (Lu et al., 2016) to compute the similarity of each node-pair and updates them by a weighted summation strategy, which performs worse than the former one.

Number of graph layer. As shown in Table 9, the model achieves the best result with 2 graph layers, and the performance will drop when the number of

Components	Module	R@1, IoU=0.5	R@1, IoU=0.7
proposal graph	w/ edge attention	46.63	26.90
	w/ edge convolution	**48.92**	**28.64**
graph layer	1 layer	47.60	27.57
	2 layers	**48.92**	**28.64**
	3 layers	48.83	28.39

Table 9: Investigation on proposal graph on ActivityNet Caption dataset.

Methods	Localization Type	R@1, IoU=0.5	R@1, IoU=0.7
SCDM	*top-down*	36.75	19.86
	ours	**43.86**	**26.42**
CMIN	*top-down*	43.40	23.88
	ours	**50.33**	**29.75**
LGI	*bottom-up*	41.51	23.07
	ours	**49.20**	**30.64**
DRN	*bottom-up*	45.45	24.36
	ours	**53.72**	**31.01**

Table 10: Our proposed adaptive proposal generation can serve as a “plug-and-play" module for existing methods. The experiments are conducted on the ActivityNet Captions dataset.

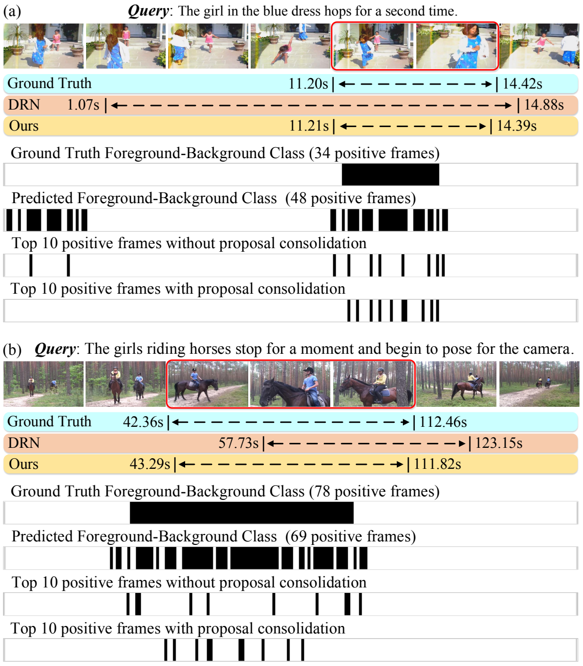


Figure 4: Typical examples of the localization results on the ActivityNet Caption dataset.

layers grows up. We give the analysis is that more graph layers will result in over-smoothing problem (Li et al., 2018) since the propagation between the nodes will be accumulated.

Plug-and-play. Our proposed adaptive proposal generation can serve as a plug-and-play for existing methods. As shown in Table 10, for *top-down* methods, we maintain their feature encoders and video-query interaction, and add the proposal generation and proposal consolidation before the localization heads. For *bottom-up* methods, we first replace their regression heads with our proposal generation process and then add the proposal consolidation process. It shows that our proposal generation and proposal consolidation can bring large improvement on both two types of methods.

4.5 Qualitative Results

To qualitatively validate the effectiveness of our APGN, we display two typical examples in Figure 4. It is challenging to accurately localize the semantic “for a second time" in the first video, because there are two separate segments corresponding to the same object “girl in the blue dress" performing the same activity “hops". For comparison, previous method DRN fails to understand the meaning of phrase “second time", and ground both two segment parts. By contrast, our method has a strong ability to distinguish these two segments in temporal dimension thanks to the positional embedding in the developed proposal graph, thus achieves more accurate localization results. Furthermore, we also display the foreground/background class of each frame in this video. With the help of the proposal consolidation module, the segment proposals of “first time" are filtered out, and all the final ranked top 10 positive frames fall in the target segment.

5 Conclusion

In this paper, we introduce APGN, a new method for temporal sentence localization in videos. Our core idea is to adaptively generates discriminative proposals and achieve both effective and efficient localization. That is, we first introduce binary classification before the boundary regression to distinguish the background frames, which helps to filter out the corresponding noisy responses. Then, the regressed boundaries on the predicted foreground frames are taken as segment proposals, which decreases a large number of poor quality proposals compared to the pre-defined ones in *top-down* framework. We further learn higher-level feature interactions between the generated proposals for refinement via a graph convolutional network. Our framework achieves state-of-the-art performance on three challenging benchmarks, demonstrating the effectiveness of our proposed APGN.

6 Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under No. 2018YFB1404102, and the National Natural Science Foundation of China under No. 61972448.

References

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. 2018. Temporally grounding natural sentence in video. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. 2020. Rethinking the bottom-up framework for query-based video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Advances in Neural Information Processing Systems (NIPS)*.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. 2020a. Reasoning step-by-step: Temporal sentence localization in videos via deep rectification-modulation network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1841–1851.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Yu Cheng, Wei Wei, Zichuan Xu, and Yulai Xie. 2021. Context-aware biaffine localizing network for temporal sentence grounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11235–11244.

Daizong Liu, Xiaoye Qu, Xiao-Yang Liu, Jianfeng Dong, Pan Zhou, and Zichuan Xu. 2020b. Jointly cross-and self-modal graph attention network for query-based moment localization. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*.

Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. 2019. DEBUG: A dense bottom-up grounding approach for natural language video localization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems (NIPS)*.

Jonghwan Mun, Minsu Cho, and Bohyung Han. 2020. Local-global video-text interactions for temporal grounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guoshun Nan, Rui Qiao, Yao Xiao, Jun Liu, Sicong Leng, Hao Zhang, and Wei Lu. 2021. Interventional video grounding with dual contrastive learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaoye Qu, Pengwei Tang, Zhikang Zou, Yu Cheng, Jianfeng Dong, Pan Zhou, and Zichuan Xu. 2020. Fine-grained iterative attention network for temporal language localization in videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4280–4288.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Cristian Rodriguez, 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 *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Joyeeta Singha, Amarjit Roy, and Rabul Hussain Laskar. 2018. Dynamic hand gesture recognition using vision-based approach for human–computer interaction. *Neural Computing and Applications*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*.

Jingwen Wang, Lin Ma, and Wenhao Jiang. 2020. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2018. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*.

Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. 2019. Multilevel language and vision integration for text-to-clip retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xun Yang, Jianfeng Dong, Yixin Cao, Xun Wang, Meng Wang, and Tat-Seng Chua. 2020. Tree-augmented cross-modal encoding for complex-query video retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1339–1348.

Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. 2021. Deconfounded video moment retrieval with causal intervention. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. 2019a. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. In *Advances in Neural Information Processing Systems (NIPS)*.

Yitian Yuan, Tao Mei, and Wenwu Zhu. 2019b. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33.

Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. 2020. Dense regression network for video grounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. 2020a. Span-based localizing network for natural language video localization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. 2020b. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. 2019. Cross-modal interaction networks for query-based moment retrieval in videos. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.