Progressive Localization Networks for Language-based Moment Localization

Qi Zheng, Jianfeng Dong, Xiaoye Qu, Xun Yang, Shouling Ji, Xun Wang, *Member, IEEE,*

***Abstract*—This paper targets the task of language-based moment localization. The language-based setting of this task allows for an open set of target activities, resulting in a large variation of the temporal lengths of video moments. Most existing methods prefer to first sample sufficient candidate moments with various temporal lengths, and then match them with the given query to determine the target moment. However, candidate moments generated with a fixed temporal granularity may be suboptimal to handle the large variation in moment lengths. To this end, we propose a novel multi-stage Progressive Localization Network (PLN) which progressively localizes the target moment in a coarse-to-fine manner. Specifically, each stage of PLN has a localization branch, and focuses on candidate moments that are generated with a specific temporal granularity. The temporal granularities of candidate moments are different across the stages. Moreover, we devise a conditional feature manipulation module and an upsampling connection to bridge the multiple localization branches. In this fashion, the later stages are able to absorb the previously learned information, thus facilitating the more fine-grained localization. Extensive experiments on three public datasets demonstrate the effectiveness of our proposed PLN for language-based moment localization and its potential for localizing short moments in long videos.**

***Index Terms*—Moment localization, Progressive Learning, Coarse-to-fine Manner, Multi-stage Model.**

I. Introduction

LOCALIZING actions/activities in videos is an increasingly important but challenging research task for video understanding, which usually can be grouped into two subfields: *i.e.,* temporal action localization [1]–[3] and language-based moment localization [4]–[6]. For temporal action localization, it aims to temporally localize segments whose action labels are within a pre-defined list of actions. Due to the pre-defined scheme and the limited action labels, it fails to effectively handle unseen activities in the real world. Therefore, a new task, language-based moment localization, is introduced by Hendricks *et al.* [7] and Gao *et al.* [8]. As shown in Figure 1a, given an untrimmed video and a natural language sentence query, the language-based moment localization task aims to temporally localize a specific video segment/moment which semantically matches the given query. This task is more flexible than temporal action localization, and recently attracts widespread interests in multiple research communities, such as computer vision [9], [10] and multimedia [11], [12]. In this paper, we target the task of language-based moment localization due to its potential applications in intelligent video surveillance, robotics, *etc* .

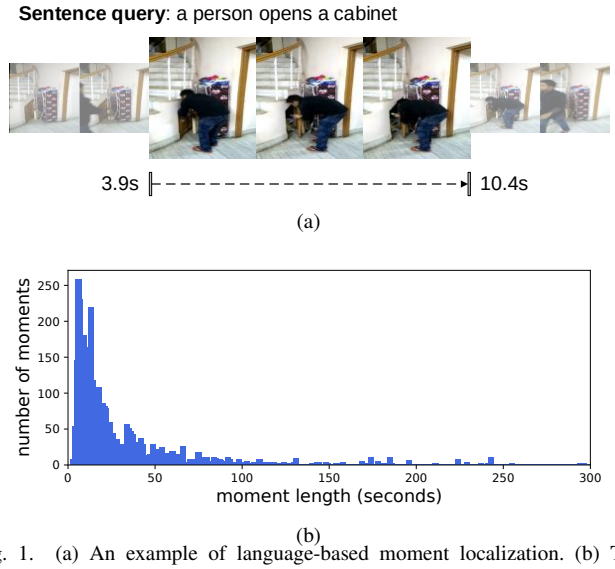


Fig. 1. (a) An example of language-based moment localization. (b) The distribution of the temporal lengths of target moments on the TACoS dataset, showing a large variation in the temporal lengths of moments.

In the language-based moment localization task, the language-based queries are usually free-form and can describe diverse video content, which allows for an open set of video activities, *e.g.,* a short activity of *person closes the door*, or a relatively more complex activity of *the person walks into the house puts down the food*. Such an open setting leads to a large variation in the temporal lengths of the target moments, as shown in Fig. 1b. Considering such a large variation, how to generate high-quality candidate moments (*a.k.a.* proposals) is one of the key questions to tackle this challenging task.

Most existing works prefer to first sample sufficient candidate moments with various temporal lengths, and then match them with the given query to determine the target moment by their cross-modal similarity [7], [8], [13]–[17]. The popular solution of generating candidate moments mainly consists of two steps. The first step is to split the input video into an ordered sequence of video clips with a fixed interval. The second step is to enumerate [7], [17] or sample [7] contiguous set of clips generated from the first step for obtaining sufficient candidate moments which have various temporal lengths. In

Q. Zheng, J. Dong and X. Wang are with the College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310035, China. E-mail: dongjf24@gmail.com

X. Qu is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Hubei 430074, China. E-mail: xiaoye@hust.edu.cn

X. Yang is with the School of Computing, National University of Singapore, Singapore 37580, Singapore. E-mail: xunyang@nus.edu.sg

S. Ji is with the School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China. E-mail: sji@zju.edu.cn

Manuscript received January XX, 2021. (Corresponding author: Jianfeng Dong)

this popular strategy, the setting of the split interval for sampling video clips is quite important to the localization performance. A large split interval usually results in a sparse set of video clips where each clip has a relatively large temporal granularity. Such video clips would construct a set of coarse-grained candidate moments that may not be able to effectively handle the target moments of short temporal lengths. In contrast, a small split interval usually results in a dense set of clips of small temporal granularity, which will accordingly generate a set of redundant candidate moments. Such redundant candidates would facilitate the localization of short-length target moments but also inevitably hinder the optimization of the learning objective.

To alleviate above dilemma, in this paper we explore the video clips of multiple diverse temporal granularities for better handling the moments with large variations in temporal lengths. We propose to progressively localize the target moment in a coarse-to-fine manner, which is inspired by humans' decision process. When humans perform language-based moment localization, they may first roughly search the target moment and then refine it. For example, given a query of *a person opens a cabinet* as illustrated in Fig. 1a, one may first search the segments where a person appears then further localize the action of opening a cabinet. To this end, we propose a Progressive Localization Network (PLN), as shown in Fig. 2, which progressively localizes the target moment via multiple stages and each stage has a localization branch with a specific temporal granularity.

Different from existing works [7], [15], [17] that utilize a fixed interval to produce video clips, we propose to use different intervals across multiple stages resulting in clips of diverse temporal granularities. Specifically, we use a large interval in the early stage and a small interval in the late stage. By this way, the localization branch in the early stage focuses on the moments generated with the clips of large temporal granularity, while the later localization branch focuses more on the moments generated with the clips of a small granularity. Such a coarse-to-fine manner contributes to handling the moments with large variations in temporal lengths. Moreover, all branches of the model are not independent as we introduce a conditional feature manipulation module and an upsampling connection to bridge them. Although the model has multiple stages, it can be trained jointly in an end-to-end style.

Our main contributions are roughly summarized as follows:

- We propose a Progressive Localization Network (PLN) which progressively localizes the target moment in a coarse-to-fine manner. Through the progressive localization via multiple stages with diverse temporal granularities, PLN shows superiority in localizing short moments in long videos. To the best of our knowledge, this paper is the first work to localize the target moment progressively.
- We propose a conditional feature manipulation that adaptively manipulates the video clip features by referring to the learned knowledge of the previous stage, making the clip features more suitable for the more fine-grained localization in the later stage. Besides, we also introduce an upsampling connection to inject the learned coarse relevance of the previous stage into the later stage. All these components are beneficial to multi-stage localization.
- We conduct extensive experiments on three public datasets: TACoS [18], ActivityNet Captions [19] and Charades-STA [8]. Our PLN achieves a new state-of-the-art performance on TACoS, and compares favorably to state-of-the-art methods on ActivityNet Captions and Charades-STA. Besides, PLN is better at localizing short moments in long videos than recent one-stage models.

II. Related Work

A. Temporal Action Localization

Given an untrimmed video, the task of temporal action localization is asked to localize segments where a pre-defined list of actions happen and predict the action label of localized segments. Owning to the success of deep learning in video understanding area, significant progress has been made in this task recently [1], [3], [20]–[24]. Existing efforts on temporal action localization could be roughly categorized into two groups: multi-stage approaches [20], [21] and one-stage approaches [24]–[26]. For multi-stage approaches, they split the localization task into multiple steps mainly involving proposal generation, classifying whether actions of interest happen in proposals, and proposal boundary refinement. For instance, Shou *et al.* [1] propose a multi-stage CNN model, which generates candidate segments, recognizes actions, and localizes temporal boundaries in three stages, respectively. By contrast, one-stage approaches integrate multiple steps together, showing much more efficient inference. It is worth noting that our multi-stage PLN is essentially different from the above multi-stage models for temporal action localization. In [20], [21], each stage plays a specific functional role and all stages have to be cascaded to generate the final localization results. By contrast, our model is able to perform localization in each stage, and the different stages focus on the distinct temporal granularities of input videos.

B. Language-based Moment Localization

As the pioneers for the language-based moment localization task, Hendricks *et al.* [7] and Gao *et al.* [8] first generate candidate moments by sliding windows, and then determine the target moment according to the given natural language query. In particular, Hendricks *et al.* [7] propose to embed candidate moments with its context and queries into a shared space, where the candidate moment showing the highest relevance score with the given query is selected as the target moment. Gao *et al.* [8] first fuse the candidate moment and query, and further employ a temporal regression network to produce their alignment scores and location offsets.

After that, this task has received increasing attention [11], [14], [15], [27]–[29]. Following the research line of [8], a number of works [13], [30] try to exploit the semantic concepts to improve the performance. Among them, instead of utilizing sliding windows to generate candidate moments, Chen *et al.* [30] employ extracted semantic concepts from the input video to generate candidate moments with high probabilities. Similarly, Xu *et al.* [31] first weight the video frames by their similarity to the given query, and select the

frames that are more relevant to the query to build candidate moments. Considering the above methods that ignore the temporal dependency of candidate moments, Zhang *et al.* [17] construct the whole candidate moments into a 2D temporal feature map and then use a convolutional network to model the temporal dependencies of adjacent moments. As such temporal dependency of candidate moments is helpful, we also inherit it into our proposed multi-stage model.

We also observe that some works utilize temporal anchors to localize the target moment without generating candidate moment in advance [4], [5], [27], [32], [33]. These methods inherit the anchor ideas of object detection [34], [35], utilizing multiple pre-defined temporal anchors of different lengths.

For instance, Chen *et al.* [32] sequentially exploit the fine-grained frame-by-word interactions between video and sentence, and feed the fused features into binary classifiers to predict relevance scores of multiple pre-defined temporal anchors at each time step. In a follow-up work, Wang *et al.* [33] additionally employ a boundary module to further explore the boundary-aware information, which leads to more precise localization. Despite the good performance, the anchors need to be carefully designed for a specific dataset and the performance is sensitive to the sizes and number of anchors [36]. Recently, we notice a solution of formulating the localization task as an end-to-end regression problem [6], [9], [10], [28], [37], [38]. For instance, Rodriguez *et al.* [6] and Mun *et al.* [10] directly regress the starting and the ending of the target moment based on the fused video-query feature by an end-to-end model. In [9], Zeng *et al.* regress the distances from each frame to the starting and ending of the target moment described by the query. With the similar idea of [9], Chen *et al.* [28] also regress the distance, while utilizes a frame feature pyramid to capture multi-level semantics.

Additionally, reinforcement learning which is well known to be effective for playing electronic games, is also applied to localize activity in videos [29], [39], [40]. These works formulate the localization task as a sequence decision problem. More recently, Cao *et al.* [12] apply adversarial learning paradigm in this task, which jointly optimize the performance of both video moment ranking and video moment localization.

As the task involves both videos and natural language, there are works that focus on devising more effective cross-modal interaction modules [9], [14]–[16], [31], [41], [42]. For example, Liu *et al.* [15] propose a word attention based on the temporal context information in videos. Yuan *et al.* [41] design an elegant co-attention module to further exploit both video and query content. Xu *et al.* [31] propose a multi-level model that integrates video and language features earlier and more tightly. Zeng *et al.* [9] devise a multi-level interaction module to fuse video and text using hierarchical feature maps. Our model is orthogonal to the improvement in cross-modal interaction, allowing us to flexibly embrace state-of-the-art cross-modal interaction modules.

Different from the above one-stage works, our paper is the first work for progressively localizing the target moment by multiple stages in a coarse-to-fine manner. Note that our multi-stage PLN is different from multi-stage model for temporal action localization [1] where Shou *et al.* generate candidate segments, recognize actions, and localize temporal boundaries in three stages respectively, and only the last stage is able to localize the target segment. By contrast, our model is able to localize moments in each stage, and each stage performs the localization in a distinct temporal granularity.

III. Progressive Localization Network

Given an untrimmed video v and a natural language sentence query s, the language-based moment localization task is required to localize a segment/moment which is semantically relevant to the given sentence query. As the task requires cross-modal reasoning and videos often contain intricate activities, it is not easy to directly localize the target moment in videos, especially for short moments in long videos. Therefore, we propose to progressively localize the target moment in a coarse-to-fine manner, and devise a model named as Progressive Localization Network (PLN). PLN has multiple stages and each stage has a localization branch (Fig. 2 illustrates the structure of two-stage PLN). In each stage $t(=1,...,T)$, a localization branch g^t predicts a new localization result $\mathbf{P^t}$ according to the given video v, the sentence query s and the learned information $\mathbf{H}^{t-1}$ of the previous stage:

$$\mathbf{P^t} = g^t(v, s, \mathbf{H}^{t-1}), \tag{1}$$

where $\mathbf{H}^{t-1}$ indicates a feature map from the localization branch of the stage $t-1$, which contains the learned coarse relevance of candidate moments with the given query. Such information benefits the later stage for more fine-grained localization and also connects the multiple localization branches in the whole model. It is worth noting that the previous information in the first stage is unavailable, so we ignore the $\mathbf{H^{t-1}}$ for g^1 and only use the video v and sentence s as the inputs. By combining the localization branches g^1 to g^T together, our proposed model localizes the target moment progressively and more accurately. In what follows, we first depict the input representation, following by the description of localization branch, conditional feature manipulation, training and inference details.

A. Input Representation

Video Representation. Following the common practice [8], [28], we use the pre-trained CNN models to represent videos. Concretely, given a video, we first split it into a fixed number of basic video units, where each unit consists of multiple consecutive frames. For each unit, we extract the deep features using a CNN model pre-trained on ImageNet per frame, and mean pooling is performed to aggregate the features. In order to make the feature more compact, we further employ a fully connected (FC) layer with a ReLU activation. Consequently, the video is described by a sequence of unit feature vectors $\mathbf{V} = \{\mathbf{u}_i\}_{i=1}^{l^v}$, where $\mathbf{u_i} \in \mathbb{R}^{d^v}$ is the feature vector of the i-th unit, d^v denotes the dimensionality of the unit feature vectors and l^v is the number of the video units. It is worth noting that 3D CNNs, such as C3D [43], can also be used for feature extraction when treating consecutive frames as individual items.

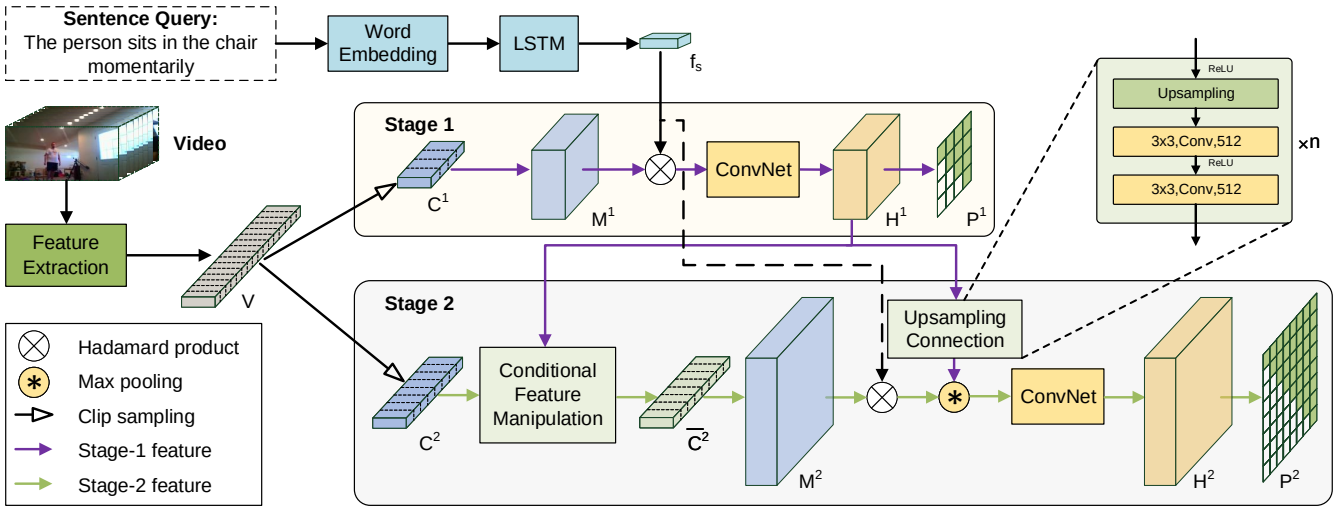


Fig. 2. The structure of our proposed two-stage Progressive Localization Network (PLN) which progressively localizes the target moment in a coarse-to-fine manner. Each stage has a localization branch, but with a distinct temporal granularity. The multiple localization branches are connected via Conditional Feature Manipulation (CFM) and Upsampling Connection (UC). The CFM adaptively recalibrates the strengths of clip features under the guidance of the previous stage, and the UC allows the learned information of the previous stage into the later stage for more fine-grained localization.

Sentence Query Representation. To represent sentence queries, we employ recurrent neural network which is known to be effective for modeling long-term word dependency in natural language text. Specifically, given a sentence query s consisting of l^s words, we first embed each word of the given sentence into a word vector space by a pre-trained GloVe model [44], which results in a sequence of word embedding vectors $\{w_1, w_2, ...w_{l^s}\}$. The word embedding vectors are then sequentially fed into a three-layer LSTM network, and the last hidden state vector of the last LSTM is taken as the representation of the sentence query, *i.e.,* $\mathbf{f}_s \in \mathbb{R}^{d^s}$, where d^s is the hidden state size of the LSTM.

B. Localization Branch

In each localization branch, there is a component localizer which aims to localize relevant video moment under the guidance of the given natural language sentence query and learned coarse information of the previous stages. Here, we choose 2D-TAN [17] as our component localizer, considering its state-of-the-art performance for language-based moment localization task. It is worth noting that theoretically other localization methods can also be used in our proposed progressive localization network. As 2D-TAN is not specifically designed for the multi-stage localization manner, it can not be directly employed. Hence, we adapt it to our proposed progressive multi-stage localization manner by additionally introducing a conditional feature manipulation and an upsampling connection layer. In what follows, we detail the localization branch in terms of candidate moment construction, relevance score prediction, and conditional feature manipulation.

Candidate Moment Construction. Given a video, this module generates candidate moments from the input video and then represent them into a 2D temporal feature map for further reasoning. Concretely, in the t-th stage, given a video represented with a sequence of video unit feature vectors $\mathbf{V} = \{\mathbf{u}_i\}_{i=1}^{l^V}$, we split the video with a specific interval thus obtaining N^t videos clips $\{\mathbf{c}_i^t\}_{i=n}^{N^t}$, denoted as $\mathbf{C}^t$, where $\mathbf{c}_i^t$ indicates the representation of i-th generated video clip obtained by mean pooling over the corresponding feature vectors. We perform the clip generation with different intervals in different stages resulting in video clips of the distinct temporal granularities, and make it satisfying $N^{t-1} < N^t$. The larger N^t means the smaller temporal granularity of generated clips. In this way, the model structure conforms to the coarse-to-fine manner and is also helpful for locating relevant moment progressively. Additionally, in the stage $t(t > 1)$, we also devise a Conditional Feature Manipulation (CFM) module to modulate the video clip features under the guidance of the previous stage, expecting to refine the clip features and make them more suitable for the more fine-grained localization in the later stage. The structure of CFM is illustrated in Fig. 4 and its detail will be introduced in the following section.

Based on the above generated video clips, we enumerate the contiguous clips to build candidate moments. Theoretically, given a sequence of N^t generated video clips, we can totally generate $\sum_{k=1}^{N^t} k$ different candidate moments with diverse lengths. In order to facilitate the exploration of the temporal dependency between candidate moments, following the previous work [17], we restructure the whole generated candidate moments to a 2D temporal feature map $\mathbf{M}^t \in \mathbb{R}^{N^t \times N^t \times d^V}$, where the first two axes indicate the start and end clip indexes at the stage t respectively, and the third axis represents the moment features. For example, $\mathbf{M}^t[i, j, :] = \mathbf{m}_{i,j}^t$ indicates the candidate moment that starting with the i-th video clip and ending with the j-th video clip, which is obtained by employing max pooling over the corresponding clip features, as exemplified in Fig.3. However, using all candidate moments will bring much more redundant information. To alleviate it, we use a sparse sampling strategy [17] that removes the redundant candidates containing large overlaps with the selected candidates. Briefly, we densely sample moments of shorts duration, and gradually increase the sampling interval when the moment duration becomes long. The restructured 2D temporal feature map $\mathbf{M}^t$ is further used for the relevance

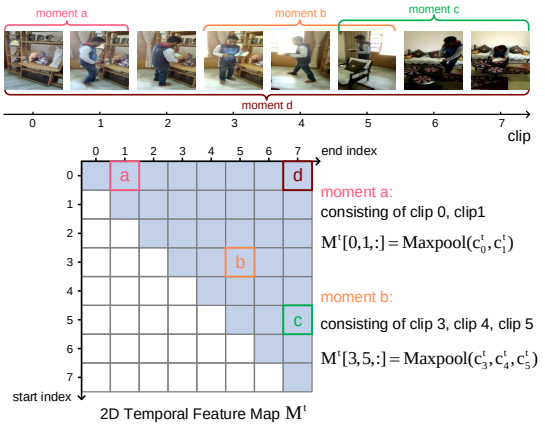


Fig. 3. An example of the 2D temporal feature map $\mathbf{M}^t$ construction on a video of eight clips. In $\mathbf{M}^t$, the first two axes indicate the start and end clip indexes respectively, and the third axis represents the moment features at the stage t. For simplicity, we omit the third axis in the figure. The white ones are invalid, as valid moment should satisfy that the start index is smaller than the end index. For the *moment a* consisting of clip 0 and clip 1, its moment feature is $\mathbf{M}^t[0,1,:]$ obtained by employing max pooling over the clip features $\mathbf{c}_0^t$ and $\mathbf{c}_1^t$.

score prediction.

Relevance Score Prediction. Given a list of candidate moments represented as a feature map $\mathbf{M}^t$ and a query represented as $\mathbf{f}_s$, this module aims to predict the relevance score map $\mathbf{P^t} \in \mathbb{R}^{N^t \times N^t}$, where $\mathbf{P^t}[i,j]$ indicates the relevance of the candidate moment starting with the i-th video clip and ending with the j-th video clip with the given query.

Concretely, we first project $\mathbf{M}^t$ and $\mathbf{f}_s$ into a d^u-dim unified space by a Fully Connected (FC) layer respectively, and fuse them through an element-wise multiplication, resulting in the fused feature map $\mathbf{F}^t$. Besides, we also consider the learned information in the previous stage crucial for the final prediction, thus employ a connection to inject the coarse relevance information of the previous stage into the follow-up stage. Notice that there is no previous stage in the first stage. Formally, after the injection, we obtain the fused feature that absorbed the information of the previous stage:

$$\mathbf{G}^t = \begin{cases} \mathbf{F}^t & \text{if t} = 1, \\ \mathbf{F}^t \circledast upsample(\mathbf{H}^{t-1}) & \text{if t} > 1, \end{cases} \tag{2}$$

where $\circledast$ denotes the element-wise max pooling, $upsample$ indicates our devised upsampling connection, and $\mathbf{H}^{t-1}$ is a feature map of the last convolutional layer in the previous stage. Here we can not directly employ a skip connection, as the sizes of $\mathbf{F}^t$ and $\mathbf{H}^{t-1}$ are different due to the different temporal granularities in different stages. Therefore, we devise a new upsampling connection which is implemented by n stacked upsampling blocks, and each block has an upsampling layer with a factor of 2 and two stacked convolutional layers with 3×3 kernels, as shown in the upper right of Fig. 2. Then a convolutional network is further utilized to exploit the candidate dependency based on the fused feature map $\mathbf{G}^t$. The convolutional network is comprised of 2 convolutional layers with 5×5 kernels, and each layer is followed by a ReLU activation. The output of the convolutional network in the t-th stage is denoted as $\mathbf{H^t}$. Finally, we feed the output $\mathbf{H^t}$ into a FC layer to predict the final relevance score map $\mathbf{P}^t \in \mathbb{R}^{N \times N}$:

$$\mathbf{P}^t = sigmoid(\mathbf{W}^t \cdot \mathbf{H}^t + \mathbf{b}^t), \tag{3}$$

where $sigmoid$ indicates an element-wise sigmoid activation, $\mathbf{W^t}$ denotes the affine matrix and $\mathbf{b^t}$ is the bias term.

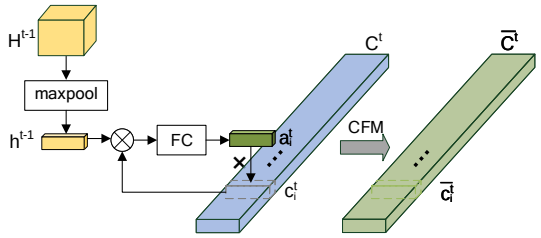


Fig. 4. The structure of conditional feature manipulation, which adaptively modulates the clip features under the guidance of the previous stage.

Conditional Feature Manipulation. As shown in Fig. 4, given a sequence of video clip feature vectors $\mathbf{C}^t = \{\mathbf{c}_i^t\}_{i=n}^{N^t}$, CFM modulates the given features under the guidance of the previous stage, obtaining the manipulated features as $\overline{\mathbf{C}^t} = \{\overline{\mathbf{c_i}^t}\}_{i=1}^{N^t}$.

Specifically, for each clip feature vector $\mathbf{c}_i^t$ in $\mathbf{C}^t$, we modulate it to $\overline{\mathbf{c}_i^t}$ with respect to the feature map $\mathbf{H}^{t-1}$. First of all, a max pooling layer is employed to aggregate $\mathbf{H}^{t-1}$ to a feature vector $\mathbf{h}^{t-1}$. Afterwards, we fuse $\mathbf{h}^{t-1}$ and $\mathbf{c}_i^t$ through element-wise multiplication and then a FC layer with the $sigmoid$ activation is employed to generate a scaling vector $\mathbf{a}_i^t \in \mathbb{R}^{d^S}$:

$$\mathbf{a}_i^t = sigmoid(\mathbf{W}_r^t \cdot (\mathbf{h}^{t-1} \odot \mathbf{c}_i) + \mathbf{b}_r^t), \tag{4}$$

where $\mathbf{W}_r^t$ and $\mathbf{b}_r^t$ parameterize the FC layer, $\odot$ indicates the Hadamard product. With the above generated scaling vector $\mathbf{a}^t$, the clip feature vector $\mathbf{v}_i$ is manipulated by scaling as:

$$\overline{\mathbf{c}_i}^t = \mathbf{c}_i \odot \mathbf{a}_i^t. \tag{5}$$

Our CFM is inspired by FiLM [45] which manipulates a neural network's intermediate features for visual reasoning. In our PLN, the motivation for CFM is two-fold: (i) we wish to adaptively recalibrate the clip features, thus making the clip features more suitable for the more fine-grained localization in the later stage. (ii) we wish to make clip features diverse across the stages thus promoting the localization branches more complementary.

After performing the conditional feature manipulation, the clips of the given video can be represented as $\overline{\mathbf{C}^t} = \{\overline{\mathbf{c_i}^t}\}_{i=1}^{N^t}$, which is used in the later stages $t(t > 1)$ to replace the original video clip features $\mathbf{C}^t$.

C. Training and Inference

Training. To train the proposed PLN, we add a classification loss per localization branch, and multiple losses of all branches are jointly considered.

Specifically, in the t-th stage, a cross-entropy loss is employed, which is defined as:

$$\mathcal{L}_t = -\frac{1}{V^t} \sum_{i=1}^{V^t} (y_i^t \log p_i^t + (1 - y_i^t) \log(1 - p_i^t)), \tag{6}$$

where $p_i^t \in \mathbf{P}^t$ is the predicted relevance score of the i-th candidate moment and V^t denotes the total number of valid candidates in the t-th stage.

For supervision labels, we employ soft labels based on candicate moments' temporal Intersection over Union (IoU) with the ground-truth moment instead of hard binary labels. Specifically, the label y_i^t in the t-th stage is defined as:

$$y_i^t = \begin{cases} 0 & o_i^t \leq \tau, \\ \frac{o_i^t - \tau}{1-\tau} & o_i^t > \tau, \end{cases} \tag{7}$$

where o_i^t denotes IoU of the corresponding candidate moment with the ground-truth moment, τ indicates the IoU threshold. Recall that candidate moments are generated with clips of different intervals in different stages, so V^t and y_i^t are different across the stages.

Finally, the joint loss of our proposed model with T stages is defined as:

$$\mathcal{L} = \sum_{t=1}^{T} \lambda_t \mathcal{L}_t, \tag{8}$$

where λ_t indicates trade-off coefficient for the t-th stage. The PLN is trained by minimizing the Eq.8 in an end-to-end manner except the CNN used for extracting video features and the GloVe for word embedding are pre-trained and fixed.

Inference.

After the model being trained, each localization branch is able to predict a relevance score map. Depending on which relevance score map is used for final localization, here we consider two strategies:

Strategy 1. In this strategy, we select one score map among all the predicted score maps, while ignoring the others. According to the selected score map, all the corresponding candidate moments will be ranked and Non Maximum Suppression (NMS) strategy is further used to remove the redundant predicted moments.

Strategy 2. Considering all the predicted relevance score maps might be valuable for localization, we ensemble all the predicted score maps to obtain the final fused score map. Specifically, if a candidate moment is associated with multiple score maps, we average their corresponding scores in all stages as the final fused relevance score. Similarly, all the corresponding candidate moments are ranked in terms of their fused score and the NMS strategy is also performed.

IV. Evaluation

In this section, we first introduce our experimental settings about datasets, evaluation metric and implementation details. Then we investigate the impact of major design choices of our proposed PLN, and compare PLN (with its best setup) against the state-of-the-art on three datasets, *i.e.,* TACoS, ActivityNet Captions, and Charades-STA. Furthermore, we conduct extensive ablation studies to verify the influence of major components in the proposed model. Finally, we analyse the influence of the moment length in this task and test the efficiency of our PLN.

A. Experimental Settings

1) Datasets: We conduct extensive experiments on three commonly used datasets: TACoS, ActivityNet Captions and Charades-STA.

TACoS [18] is built based on the MPII Cooking Composite Activities video dataset [46], consisting of 127 videos about activities that happened in the kitchen. The average length of videos is about 286 seconds, and the average length of target moments is about 27 seconds. Among them, lots of queries are about short fine-grained actions, such as *takes out the knife*. Such short spanned moments show the challenging nature of this dataset. We use the standard split, *i.e.,* 10146, 4589 and 4083 sentence-moment pairs for training, validation and testing, respectively.

ActivityNet Captions [19] is the largest dataset for the task of moment localization with natural language, which is built on ActivityNet v1.3 dataset [47]. The dataset is originally developed for video captioning, and now popular for the task of moment localization with natural language as these two tasks are reversible. Compared with the above two datasets, the videos are open domain and more diverse in content. It totally has $19,209$ videos, and the average length is about 117 seconds. For the fair comparison, we follow the dataset split used in [17], [27] using 37417, 17505, and 17031 sentence-moment pairs for training, validation, and testing, respectively.

Charades-STA [8] is built on the top of the Charades dataset [48] which is introduced for action recognition and localization. The Charades dataset has $9,848$ videos about daily indoor activities and each video is annotated with the video-level descriptions. The videos are 30 seconds long on average. Gao *et al.* [8] extend this by including the sentence-level temporal annotation to create the Charades-STA dataset which has 12408 sentence-moment pairs for training and 3720 pairs for testing.

2) Evaluation Metric: Following the previous work [32], we report the performance in terms of the metrics $Rank@n, IoU = m$ ($n \in \{1, 5\}, m \in \{0.1, 0.3, 0.5, 0.7\}$) and $mIoU$. $Rank@n, IoU = m$ is defined as the percentage of test queries for which at least one relevant moment with higher IoU than m is found among the top-n retrieved results. $mIoU$ is the mean IoU of top-1 prediction result with the ground truth moment over all test queries. All results are reported in percentage (%).

3) Implementation Details: Here we detail common implementations of our proposed model in terms of model structure, training and inference. For the sentence encoding, the size of hidden states in the LSTM is 512, so $d^s = 512$. For the video features, we utilize the same visual features provided by [17], *i.e.,* 4096-dim C3D feature [43] on TACoS, 500-dim C3D feature on ActivityNet Captions, and 4096-dim VGG feature [49] on Charades-STA. Besides, we notice a trend of using a stronger I3D feature on Charades-STA, so we also use the 1024-dim I3D feature provided by [6]. The parameters of the convolution networks (ConvNet) in all stages are shared. For the upsampling connection, we use 2 upsampling blocks for two-stage models(*i.e.,* n=2), and 1 upsampling blocks for three-stage and four-stage models(*i.e.,* n=1). We use two-stage PLN to compare the state-of-the-art models, and the number

TABLE I
PERFORMANCE OF PLN USING DIFFERENT PREDICTION STRATEGIES ON THE TACOS DATASET.

Method	Rank@1,IoU=m			Rank@5,IoU=m			mIoU
	0.3	0.5	0.7	0.3	0.5	0.7	
two-stage:							
Strategy 1 (t=1)	37.19	22.84	9.87	60.11	42.24	18.30	24.63
Strategy 1 (t=2)	**43.89**	**31.12**	**16.10**	65.11	**52.89**	**27.52**	**29.70**
Strategy 2	43.59	30.89	15.87	**65.28**	52.54	27.44	29.43
three-stage:							
Strategy 1 (t=1)	34.29	20.72	7.62	58.36	39.49	15.30	22.75
Strategy 1 (t=2)	37.97	26.37	11.52	63.13	49.34	22.72	25.79
Strategy 1 (t=3)	**39.54**	**28.57**	**13.35**	**64.01**	49.86	**24.12**	**26.89**
Strategy 2	39.22	28.29	12.87	63.88	**50.24**	22.94	26.54

t denotes which stage is selected for strategy 1. For example, t=1 means that the predicted relevance score map of the first stage is used for prediction.

TABLE II
PERFORMANCE OF PLN WITH THE DIFFERENT NUMBER OF STAGES ON THE TACOS DATASET.

Method	Rank@1,IoU=m			Rank@5,IoU=m			mIoU
	0.3	0.5	0.7	0.3	0.5	0.7	
one-stage:							
N^t=16	30.93	17.22	7.03	55.52	33.54	12.31	20.24
N^t=32	35.04	20.54	8.42	58.66	39.62	16.40	22.92
N^t=64	37.77	22.92	10.52	60.36	45.26	20.99	24.79
N^t=128	36.32	23.82	11.42	60.96	46.59	23.64	24.51
two-stage:							
N^t=32-128	**43.89**	**31.12**	**16.10**	**65.11**	**52.89**	**27.52**	**29.70**
three-stage:							
N^t=32-64-128	39.54	28.57	13.35	64.01	49.86	24.12	26.89
four-stage:							
N^t=16-32-64-128	38.42	25.17	11.65	63.63	46.96	22.99	25.44

Numbers separated with a hyphen '-' denote the numbers of generated clips used in each stage. Two-stage PLN strikes the best balance between model capacity and generalization ability.

TABLE III
EFFECTIVENESS OF THE COARSE-TO-FINE MANNER ON TACOS.

Method	Rank@1,IoU=m			Rank@5,IoU=m			mIoU
	0.3	0.5	0.7	0.3	0.5	0.7	
N^t=64	**37.77**	22.92	10.52	60.36	45.26	20.99	24.79
N^t=32-32	34.54	20.62	7.72	59.94	41.31	17.15	22.63
N^t=64-64	35.67	21.07	9.20	59.29	43.64	20.64	23.00
N^t=32-64*	37.29	**23.69**	**11.12**	**61.93**	**45.69**	**21.19**	**25.01**
N^t=128	36.32	23.82	11.42	60.96	46.59	23.64	24.51
N^t=32-32	34.54	20.62	7.72	59.94	41.31	17.15	22.63
N^t=128-128	33.34	19.65	9.55	58.69	43.44	22.57	22.22
N^t=32-128*	**43.89**	**31.12**	**16.10**	**65.11**	**52.89**	**27.52**	**29.70**

* denotes the coarse-to-fine manner.

of sampled clips N^t is set to 32 and 128 on TACoS, 16 and 64 on ActivityNet Captions and Charades-STA.

On ActivityNet Captions, we additionally add a sinusoidal positional encoding [50] to a sequence unit features, thus boost the order of the sequence. For the model training, we empirically set the loss weights λ_t to 1.0 and 1.5 for the two-stage model, 1.0, 1.3 and 1.5 for the three-stage model, and 1.0, 1.2, 1.5 and 2.0 for the four-stage model. The IoU threshold τ with ground-truth moment is set to be 0.5. Additionally, we use Adam optimizer [51] to train the model. The initial learning rate and the batch size are empirically set to be 0.0001 and 32 respectively. The maximum number of epochs is 50. For model inference, we use non maximum suppression (NMS) strategy to remove the redundant predicted moments. Following the previous work [17], the threshold of NMS is set to be 0.4 on TACoS, 0.5 on ActivityNet Captions and 0.45 on the Charades-STA dataset.

B. Properties of PLN

Before comparing with the state-of-the-art methods, we first investigate the impact of major design choices, *i.e., Which prediction strategy?*, *How many stages?* to better understand our proposed PLN on the TACoS dataset.

1) Which Prediction Strategy?: In this experiment, we use the two-stage and three-stage PLN as our base models to investigate the two different prediction strategies. Table I summarizes the performance. For the Strategy 1, models using the score map of the later stage achieve better performance. For example, for the three-stage model, its $mIoU$ score gradually improves from 22.75, 25.79, to 26.89 as the stage increases.

Additionally, we observe that in general the Strategy 2 fusing the score maps of all stages is slightly worse than the Strategy 1 using the last stage. This result is inconsistent with that the fusion typically gains some performance improvement in other multimedia related tasks [52]. We attribute it to that the later stage in PLN has already absorbed the information of the previous stage, thus further fusing the results of previous stages has less impact. Therefore, we use Strategy 1 based on the output of the last stage as our inference strategy.

2) How Many Stages?: Table II summarizes the performance of PLN variants with the different number of stages.

The models with multiple stages beat the one-stage counterparts by a clear margin. The result shows the effectiveness of localizing moments by multiple stages. Among multi-stage models, the two-stage one turns out to be the most effective, but using more stages has no performance gain while worsening the performance. While its learning capacity increases as the stage of the model increases, the chance of over-fitting also increases. Overall the two-stage PLN strikes the best balance between model capacity and generalization ability, so we use the two-stage PLN to compare to state-of-the-art models in the following.

3) Coarse-to-Fine Manner?: To verify the viability of our devised coarse-to-fine manner for PLN, we compare it with the PLN using the same temporal granularity of video clips in all stages (N^t is the same for all stages). As shown in Table III, our PLN with the coarse-to-fine temporal granularities (line 4 in each table block) outperforms the model with the same temporal granularity across stages (line 2, 3), showing the superiority of the coarse-to-fine manner for the multi-stage PLN. Additionally, we also report the results of one-stage model (line 1), and find it better than the two-stage model without the coarse-to-fine temporal granularities (line 3).

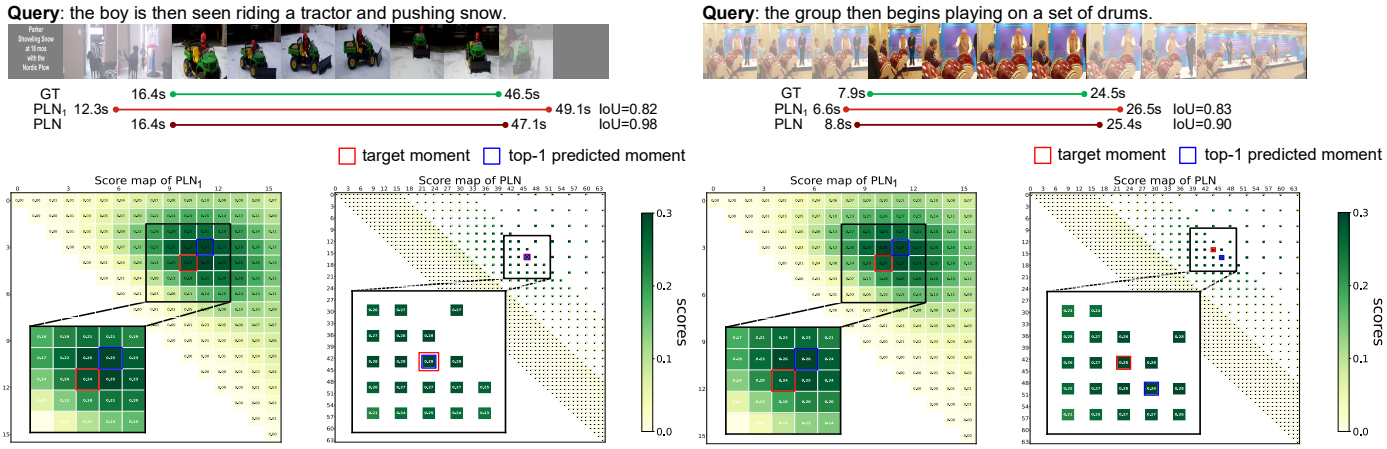
The result reveals the necessity of the coarse-to-fine manner for multi-stage localization.

C. Comparison to the State-of-the-Arts

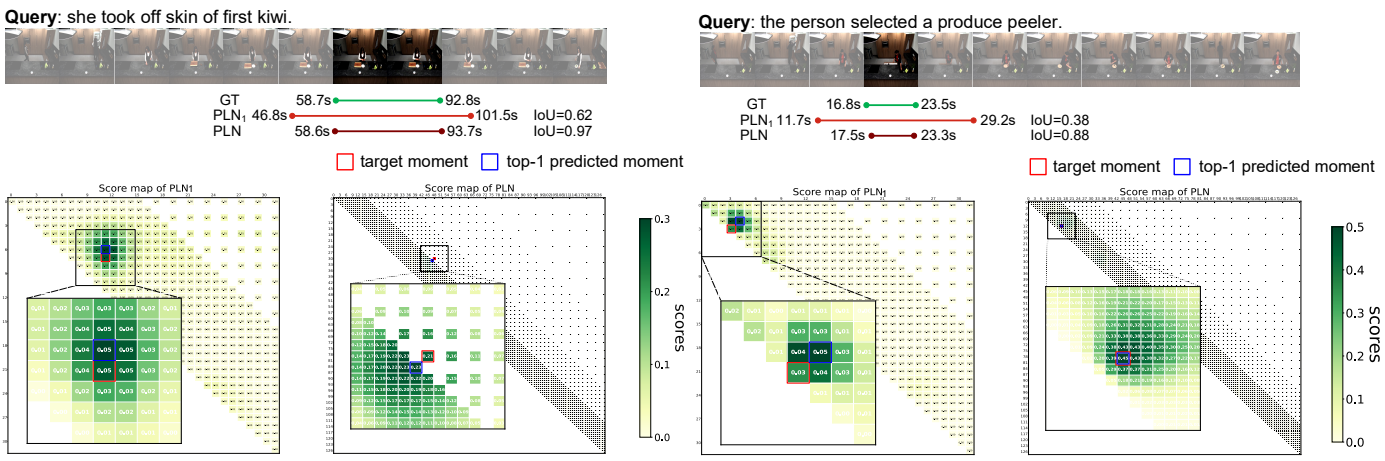
Table IV summarizes the performance comparison on both TACoS and ActivityNet Captions, where all the models use the same C3D feature. Besides, except for our proposed PLN, all other models are one-stage. Our proposed multi-stage PLN consistently achieves the best performance on TaCoS, and compares favorably to existing methods on ActivityNet

TABLE IV
PERFORMANCE COMPARISON ON TACOS AND ACTIVITYNET CAPTIONS. ALL THE MODELS USE THE SAME C3D FEATURE, AND THE PERFORMANCE SCORES ARE DIRECTLY CITED FROM THE ORIGINAL PAPERS.

Method	Source	TACoS							ActivityNet Captions						
		Rank@1,IoU=m			Rank@5,IoU=m			mIoU	Rank@1,IoU=m			Rank@5,IoU=m			mIoU
		0.1	0.3	0.5	0.1	0.3	0.5		0.3	0.5	0.7	0.3	0.5	0.7	
CTRL [7]	ICCV 2017	24.32	18.32	13.30	48.73	36.69	25.42	11.98	-	-	-	-	-	-	-
ACRN [14]	SIGIR 2018	24.22	19.52	14.62	47.42	34.97	24.88	-	49.70	31.67	11.25	76.50	60.34	38.57	-
TGN [32]	EMNLP 2019	41.87	21.77	18.90	53.40	39.06	31.02	17.93	-	-	-	-	-	-	-
ACL-K [13]	WACV 2019	31.64	24.17	20.01	57.85	42.15	30.66	-	-	-	-	-	-	-	-
CMIN [27]	SIGIR 2019	32.48	24.64	18.05	62.13	38.46	27.02	-	**63.61**	43.40	23.88	80.54	67.95	50.73	-
DEBUG [37]	EMNLP 2019	41.15	23.45	-	-	-	-	16.03	55.91	39.72	-	-	-	-	39.51
TripNet [40]	arXiv 2019	-	23.95	19.17	-	-	-	-	48.42	32.19	13.93	-	-	-	-
ABLR [41]	AAAI 2019	34.70	19.50	9.40	-	-	-	13.40	55.67	36.79	-	-	-	-	36.99
QSPN-Cap [31]	AAAI 2019	-	-	-	-	-	-	-	45.30	27.70	13.60	75.70	59.20	38.30	-
SLTA [16]	ICMR 2019	23.13	17.07	11.92	46.52	32.90	20.86	-	-	-	-	-	-	-	-
SCDM [4]	NeuralPS 2019	-	26.11	21.17	-	40.16	32.18	-	54.80	36.75	19.86	77.29	64.99	41.53	-
VSLNet [38]	ACL 2020	-	29.61	24.27	-	-	-	24.11	63.16	43.22	26.16	-	-	-	43.19
CBP [33]	AAAI 2020	-	27.31	24.79	-	43.64	37.40	21.59	54.30	35.76	17.80	77.63	65.89	46.20	36.85
GDP [28]	AAAI 2020	39.68	24.14	-	-	-	-	16.18	56.17	39.27	-	-	-	-	39.8
TSP-PRL [29]	AAAI 2020	-	-	-	-	-	-	-	56.08	38.76	-	-	-	-	39.21
2D-TAN [17]	AAAI 2020	47.59	37.29	25.32	74.46	57.81	45.04	25.19	59.45	44.51	27.38	85.65	77.13	62.26	43.29
PMI-LOC [53]	ECCV 2020	-	-	-	-	-	-	-	59.69	38.28	17.83	-	-	-	-
DRN [9]	CVPR 2020	-	-	23.17	-	-	33.36	-	-	45.45	24.36	-	**77.97**	50.30	-
RBM [10]	CVPR 2020	-	-	-	-	-	-	-	58.52	41.51	23.07	-	-	-	41.13
PLN	This work	**53.74**	**43.89**	**31.12**	**75.56**	**65.11**	**52.89**	**29.70**	59.65	**45.66**	**29.28**	**85.66**	76.65	**63.06**	**44.12**



(a) on ActivityNet Captions



(b) on TACoS

Fig. 5. Selected examples of language-based moment localization by our model on ActivityNet Captions and TACoS. For each query and the corresponding video, three localization results are shown. GT denotes the ground-truth result. PLN_1 indicates our proposed two-stage PLN using the output of stage 1 for prediction, while PLN uses the output of stage 2. Below the localization results are the predicted score maps of PLN_1 and PLN, where red boxes indicate target moments and blue boxes indicate top-1 predicted moments. The later stage of PLN gives more accurate localization. Best viewed in zoom in.

TABLE V
PERFORMANCE COMPARISON WITH THE SAME VGG OR I3D FEATURE ON CHARADES-STA.

Method	Source	Rank@1,IoU=m			Rank@5,IoU=m			mIoU
		0.3	0.5	0.7	0.3	0.5	0.7	
VGG:								
SAP [30]	AAAI 2019	-	27.42	13.36	-	66.37	38.15	-
SM-RL [39]	CVPR 2019	-	24.36	11.17	-	61.25	32.08	-
2D-TAN [17]	AAAI 2020	57.31	42.80	23.25	93.49	83.84	54.17	39.23
DRN [9]	CVPR 2020	-	42.90	23.68	-	**87.80**	54.87	-
PLN	This work	**59.33**	**45.43**	**26.26**	**95.48**	86.32	**57.02**	**41.28**
I3D:								
MAN [5]	CVPR 2019	-	46.53	22.72	-	86.23	53.72	-
SCDM [4]	NeuralPS 2019	-	54.44	33.43	-	74.43	58.08	-
PfTML [6]	WACV 2020	67.53	52.02	33.74	-	-	-	-
DRN [9]	CVPR 2020	-	53.09	31.75	-	**89.06**	60.05	-
RBM [10]	CVPR 2020	**72.96**	**59.46**	**35.48**	-	-	-	**51.38**
PLN	This work	68.60	56.02	35.16	**94.54**	87.63	**62.34**	49.09

TABLE VI
ABLATION STUDIES OVER THE TWO-STAGE PLN. OUR FULL MODEL PERFORMS THE BEST ON THREE DATASETS.

Dataset	Method	Rank@1,IoU=m			mIoU
		0.3	0.5	0.7	
Charades-STA	Full model	**59.33**	**45.43**	**26.26**	**41.28**
	w/o CFM	57.50	44.09	25.89	40.19
	w/o UC	56.10	43.52	26.40	39.23
	w/o UC and CFM	56.13	42.69	24.81	38.90
TaCoS	Full model	**43.89**	**31.12**	**16.10**	**29.70**
	w/o CFM	43.01	30.97	15.22	29.05
	w/o UC	36.09	22.44	10.77	23.82
	w/o UC and CFM	35.04	25.37	12.12	23.92
ActivityNet Captions	Full model	**59.65**	**45.66**	**29.28**	**44.12**
	w/o CFM	58.99	44.61	28.44	43.71
	w/o UC	59.19	44.52	27.97	43.39
	w/o UC and CFM	57.21	43.23	27.55	42.34

Captions. Moreover, our multi-stage PLN in a coarse-to-fine manner outperforms 2D-TAN with a clear margin. 2D-TAN is the most state-of-the-art method on both datasets, which can be deemed as the degraded version of our proposed model, only has one localization branch with clips of a fixed temporal granularity as input. The results again justify the viability of localizing the target moment via multiple stages with diverse temporal granularities. Additionally, we also notice a phenomenon that the performance improvement of our PLN over 2D-TAN is more significant on TACoS than that on ActivityNet Captions. As the moment length proportion of the entire video on TACoS is much smaller than that on ActivityNet Captions, we conclude that our PLN with multiple stages is more beneficial for localizing relatively short moments in long videos.

The performance comparison on Charades-STA is shown in Table V. With the same VGG feature, our PLN performs the best except in terms of Rank@5 with IoU of 0.5. The result again confirms the effectiveness of PLN. With the same I3D feature, PLN is slightly worse than RBM. We attribute it to the fact that RBM explores local-global video-text interactions via temporal attention for video-text fusion, while PLN utilizes a relatively simple element-wise multiplication. However, if videos are too long, the temporal attention typically fails to work well. It has been verified by the result that RBM is consistently worse than PLN on ActivityNet Captions where the average length of videos is much longer than that on Charades-STA (117 seconds vs. 30 seconds).

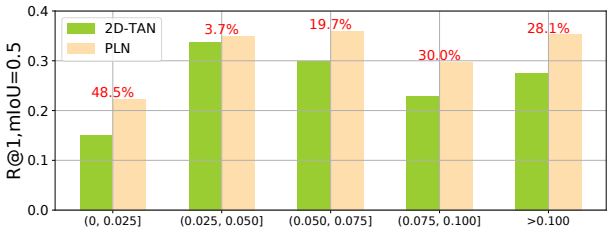
Fig. 5 shows some localization examples returned by our proposed two-stage PLN on ActivityNet Captions. Note that PLN uses the predicted score map of the last stage, while the PLN variant, PLN_1, utilizes the score map of the first stage. In general, PLN gives more accurate localization than PLN_1. For example, in the first example, PLN achieves a high IoU score of 0.98, while PLN_1 gives a score of 0.82. It is worth noting that PLN fuses both coarse and fine-grained information of all stages for localization. The results show the importance of the coarse-to-fine information for language-based moment localization. Additionally, we also illustrate the predicted score map of PLN_1 and PLN. In score maps, each score indicates its predicted relevance of a specific candidate moment with respect to the given query. Besides, candidate moments that are closer to each other in the score map typically have larger IoU between each other. As shown in the predicted score maps of our PLN, moments near the target moment give high scores, while moments far from the target moment have relatively low scores. The results show that our model is able to distinguish which moment is more relevant to the given query.

D. Ablation Studies.

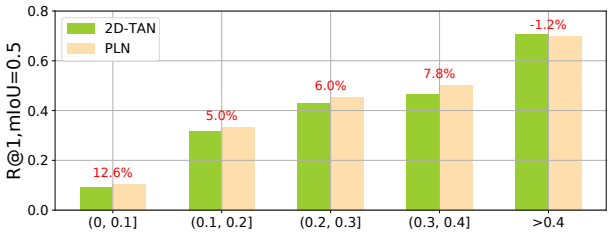
In order to investigate the contribution of each component in the proposed PLN, we conduct ablation studies and summarize the results in Table VI. On Charades-STA, we observe that our full model performs the best in terms of all metrics. Removing each component from PLN, *i.e.,* upsampling connection (UC), or conditional feature manipulation (CFM), would result in performance degeneration. The result demonstrates the effectiveness of each component in our PLN. Besides, the model without both UC and CFM performs the worst. Recall that the localization branch of different stages in this model are independent, and the information of the previous stage is not fed to the later stage. It not only reflects the complementary of UC and CFM, but also shows that bridging the multiple localization branches of distinct stages is important for multi-stage localization models. On the TACoS dataset and ActivityNet Captions dataset, our full model also outperforms the other degraded counterparts.

E. Analysis on Different Types of Moments

To investigate how our multi-stage PLN performs on different groups of moments, we group target moments according to their length proportion of the entire video length in the test set. We compare PLN with the 2D-TAN model that to some extent can be seemed as a one-stage PLN. As shown in Fig. 6, we observe a phenomenon that the both models perform the worst on the first group where moments have the shortest average temporal duration. It to some extent demonstrates that localizing the short moments is more challenging. In almost groups, our PLN consistently outperforms 2D-TAN, which again indicates the effectiveness of our PLN for language-based moment localization. Especially, PLN achieves the highest relative performance improvement of 48.5% in the first group on TACoS and 12.6% on ActivityNet Captions. It demonstrates that our proposed multi-stage localization mechanism with the coarse-to-fine manner can better handle



(a)



(b)

Fig. 6. Detailed performance comparison between 2D-TAN and our proposed PLN on (a) TACoS and (b) ActivityNet Captions. Moments have been grouped in terms of their length proportion in the entire video. The red number over the bins is the relative performance improvement of PLN over 2D-TAN.

TABLE VII
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE ACTIVITYNET CAPTIONS DATASET IN TERMS OF MODEL SIZE AND COMPUTATION OVERHEAD AT THE INFERENCE STAGE.

Model	Source	Model Complexity	
		Parameters (M) ↓	FLOPs (G) ↓
CMIN [27]	SIGIR 2019	146.00	17.30
2D-TAN [17]	AAAI 2020	60.93	104.52
DRN [9]	CVPR 2020	162.62	**11.60**
RBM [10]	CVPR 2020	155.65	15.52
PLN	This work	**44.41**	124.30

Although our PLN takes more FLOPs than other models, it takes approximately 0.025 seconds to localize a relevant moment from a video. The localization speed is adequate for instant response.

the relatively short target moments in long videos than one-stage 2D-TAN.

F. Efficiency Test

In this experiment, we compare our PLN with recently proposed models in terms of model size and computation overhead at the inference stage. Here we choose CMIN [27], 2D-TAN [17], DRN [9], RBM [10], considering their state-of-the-art performance and code available. For each model, we measure the number of FLOPs it takes to perform localization on a video of 256 sampled frames with respect to a textual query of 10 words. The comparison results are summarized in Table VII, which are tested on a normal computer with 128G RAM and a GTX TITAN Xp GPU. In terms of the model size, our PLN and 2D-TAN are more lightweight than others. It is mainly due to both PLN and 2D-TAN employ element-wise multiplication for video-text fusion, while CMIN, DRN and RBM use relative more complex video-text interactions. It is worth noting that 2D-TAN to some extent can be regarded as a one-stage PLN, but PLN has less trainable parameters. We attribute it to two reasons. Firstly, PLN only uses two stacked convolutional layers in ConvNet, while 2D-TAN uses eight stacked convolutional layers in ConvNet. Secondly, the parameters of ConvNet of different stages in PLN are shared. Additionally, our PLN unsurprisingly takes more FLOPs than other models, as it consists of multiple localization branches. Although the multi-stage manner of PLN takes more FLOPs, it brings stability and improvement in performance. We also evaluat the speed of PLN. It takes approximately 0.025 seconds to localize a relevant moment from a video, which is adequate for instant response.

V. CONCLUSION

This paper shows the viability of resolving language-based moment localization in a progressive coarse-to-fine manner. We contribute a novel multi-stage Progressive Localization Network which is capable of localizing the target moment progressively via multiple localization branches. The localization branches are connected via a conditional feature manipulation module and an upsampling connection, making the later stage absorb the previously learned coarse information, thus facilitate the more fine-grained localization. Moreover, we believe that this simple and effective multi-stage progressive architecture can be of interest to many language-based moment localization research efforts. Although our PLN takes more FLOPs than recently proposed models, its localization speed is adequate for instant response. In future work, we will further accelerate our proposed model, and apply the idea of coarse-to-fine localization to other related tasks such as temporal action localization.

ACKNOWLEDGMENT

This work was supported by National Key R&D Program of China (No. 2018YFB1404102), NSFC (No. 61902347), ZJNSF (No. LQ19F020002, LY21F020010), the Research Program of Zhejiang Lab (No. 2019KD0AC02), and the Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

REFERENCES

[1] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[2] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3628–3636.

[3] P. Chen, C. Gan, G. Shen, W. Huang, R. Zeng, and M. Tan, "Relation attention for temporal action localization," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2723–2733, 2020.

[4] Y. Yuan, L. Ma, J. Wang, W. Liu, and W. Zhu, "Semantic conditioned dynamic modulation for temporal sentence grounding in videos," in *Advances in Neural Information Processing Systems*, 2019, pp. 536–546.

[5] D. Zhang, X. Dai, X. Wang, Y.-F. Wang, and L. S. Davis, "MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1247–1257.

[6] C. Rodriguez, E. Marrese-Taylor, F. S. Saleh, H. LI, and S. Gould, "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*, 2020, pp. 2464–2473.

[7] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with natural language," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5803–5812.

[8] J. Gao, C. Sun, Z. Yang, and R. Nevatia, "TALL: Temporal activity localization via language query," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5267–5275.

[9] R. Zeng, H. Xu, W. Huang, P. Chen, M. Tan, and C. Gan, "Dense regression network for video grounding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 287–10 296.

[10] J. Mun, M. Cho, and B. Han, "Local-global video-text interactions for temporal grounding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 810–10 819.

[11] S. Zhang, J. Su, and J. Luo, "Exploiting temporal relationships in video moment localization with natural language," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1230–1238.

[12] D. Cao, Y. Zeng, X. Wei, L. Nie, R. Hong, and Z. Qin, "Adversarial video moment retrieval by jointly modeling ranking and localization," in *Proceedings of the ACM International Conference on Multimedia*, 2020, pp. 898–906.

[13] R. Ge, J. Gao, K. Chen, and R. Nevatia, "MAC: Mining activity concepts for language-based temporal localization," in *IEEE Winter Conference on Applications of Computer Vision*, 2019, pp. 245–253.

[14] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua, "Attentive moment retrieval in videos," in *The 41st international ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 15–24.

[15] M. Liu, X. Wang, L. Nie, Q. Tian, B. Chen, and T.-S. Chua, "Cross-modal moment localization in videos," in *Proceedings of the ACM International Conference on Multimedia*, 2018, pp. 843–851.

[16] B. Jiang, X. Huang, C. Yang, and J. Yuan, "Cross-modal video moment retrieval with spatial and language-temporal attention," in *Proceedings of the International Conference on Multimedia Retrieval*, 2019, pp. 217–225.

[17] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2d temporal adjacent networks for moment localization with natural language," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 870–12 877.

[18] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal, "Grounding action descriptions in videos," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 25–36, 2013.

[19] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, "Dense-captioning events in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 706–715.

[20] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7094–7103.

[21] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1130–1139.

[22] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5734–5743.

[23] Q. Liu and Z. Wang, "Progressive boundary refinement network for temporal action detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 11 612–11 619.

[24] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "SST: Single-stream temporal action proposals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2911–2920.

[25] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 25th ACM international Conference on Multimedia*, 2017, pp. 988–996.

[26] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," in *British Machine Vision Conference*, 2017.

[27] Z. Zhang, Z. Lin, Z. Zhao, and Z. Xiao, "Cross-modal interaction networks for query-based moment retrieval in videos," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 655–664.

[28] L. Chen, C. Lu, S. Tang, J. Xiao, D. Zhang, C. Tan, and X. Li, "Rethinking the bottom-up framework for query-based video localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 10 551–10 558.

[29] J. Wu, G. Li, S. Liu, and L. Lin, "Tree-structured policy based progressive reinforcement learning for temporally language grounding in video," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 12 386–12 393.

[30] S. Chen and Y.-G. Jiang, "Semantic proposal for activity localization in videos via sentence query," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 8199–8206.

[31] H. Xu, K. He, B. A. Plummer, L. Sigal, S. Sclaroff, and K. Saenko, "Multilevel language and vision integration for text-to-clip retrieval." in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 9062–9069.

[32] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua, "Temporally grounding natural sentence in video," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 162–171.

[33] J. Wang, L. Ma, and W. Jiang, "Temporally grounding language queries in videos by contextual boundary-aware prediction." in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 12 168–12 175.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[35] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[36] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9627–9636.

[37] C. Lu, L. Chen, C. Tan, X. Li, and J. Xiao, "Debug: A dense bottom-up grounding approach for natural language video localization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 2019, pp. 5147–5156.

[38] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, "Span-based localizing network for natural language video localization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 6543–6554.

[39] W. Wang, Y. Huang, and L. Wang, "Language-driven temporal activity localization: A semantic matching reinforcement learning model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 334–343.

[40] M. Hahn, A. Kadav, J. M. Rehg, and H. P. Graf, "Tripping through time: Efficient localization of activities in videos," in *British Machine Vision Conference*, 2020.

[41] Y. Yuan, T. Mei, and W. Zhu, "To find where you talk: Temporal sentence localization in video with attention based location regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 9159–9166.

[42] S. Chen and Y.-G. Jiang, "Hierarchical visual-textual graph for temporal activity localization via language," in *European Conference on Computer Vision*, 2020, pp. 601–618.

[43] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[44] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[45] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[46] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, "Script data for attribute-based recognition of composite activities," in *European Conference on Computer Vision*, 2012, pp. 144–157.

[47] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[48] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *European Conference on Computer Vision*, 2016, pp. 510–526.

[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[52] J. Dong, X. Li, and D. Xu, "Cross-media similarity evaluation for web image retrieval in the wild," *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2371–2384, 2018.
[53] S. Chen, W. Jiang, W. Liu, and Y.-G. Jiang, "Learning modality interaction for temporal sentence localization and event captioning in videos," in *European Conference on Computer Vision*, 2020, pp. 333–351.