Dense Regression Network for Video Grounding

Runhao Zeng[1,3*] Haoming Xu[1] Wenbing Huang[4] Peihao Chen[1] Mingkui Tan[1†] Chuang Gan[2]
[1]School of Software Engineering, South China University of Technology, China
[2]MIT-IBM Watson AI Lab [3]Peng Cheng Laboratory, Shenzhen, China
[4]Beijing National Research Center for Information Science and Technology (BNRist),
Department of Computer Science and Technology, Tsinghua University
{runhaozeng.cs, ganchuang1990}@gmail.com, hwenbing@126.com, mingkuitan@scut.edu.cn

Abstract

We address the problem of video grounding from natural language queries. The key challenge in this task is that one training video might only contain a few annotated starting/ending frames that can be used as positive examples for model training. Most conventional approaches directly train a binary classifier using such imbalance data, thus achieving inferior results. The key idea of this paper is to use the distances between the frame within the ground truth and the starting (ending) frame as dense supervisions to improve the video grounding accuracy. Specifically, we design a novel dense regression network (DRN) to regress the distances from each frame to the starting (ending) frame of the video segment described by the query. We also propose a simple but effective IoU regression head module to explicitly consider the localization quality of the grounding results (i.e., the IoU between the predicted location and the ground truth). Experimental results show that our approach significantly outperforms state-of-the-arts on three datasets (i.e., Charades-STA, ActivityNet-Captions, and TACoS).

1. Introduction

Video grounding is an important yet challenging task in computer vision, which requires the machine to watch a video and localize the starting and ending time of the target video segment that corresponds to the given query, as shown in Figure 1. This task has drawn increasing attention over the past few years due to its vast potential applications in video understanding [38, 3, 45, 44, 6], video retrieval [42, 8], and human-computer interaction [35, 20, 50], etc.

The task, however, is very challenging due to several reasons: 1) It is nontrivial to build connections between the query and complex video contents; 2) Localizing actions of interest precisely in a video with complex backgrounds is very difficult. More critically, a video can often contain many thousands of frames, but it may have only a few annotated starting/ending frames (namely the positive training examples), making the problem even more challenging. Previous approaches often adopt a two-stage pipeline [9, 40, 10], where they generate the proposals and rank them according to their similarities with the query. However, this pipeline incurs two issues: 1) One video often contains thousands of proposals, resulting in a heavy computation cost when comparing proposal-query pairs. 2) The performance highly relies on the quality of proposals. To address the above issues, one-stage video grounding methods [5, 43, 11] have been studied. Yuan *et al*. [43] propose to learn a representation of the video-query pair and use a multi-layer perceptron (MLP) to regress the starting and ending time. Chen *et al*. [5] and Ghosh *et al*. [11] attempt to predict two probabilities at each frame, which indicate whether this frame is a starting (or ending) frame of the target video segment. The grounding result is obtained by selecting the frame with the largest starting (or ending) probability. However, the existing two-stage and one-stage methods have one common issue: they neglect the rich information from the frames within the ground truth.

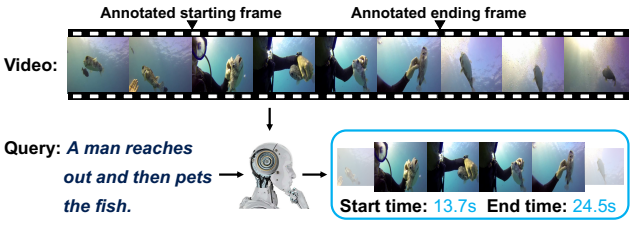

Figure 1. An illustrative example of the video grounding task. Given a video and a query in natural language, the video grounding task aims to identify the starting time and the ending time of the video segment described by the query. One key challenge of this task is how to leverage dense supervision upon sparsely annotated starting and ending frames only.

*This work was done when Runhao Zeng was a research intern at Peng Cheng Laboratory, Shenzhen, China.
†Corresponding author

Recently, anchor-free approaches [21, 31, 36, 26, 24] for one-stage object detection become increasingly popular because of their simplicity and effectiveness. In this vein, Tian *et al*. [36] propose the FCOS framework to solve object detection in a per-pixel prediction fashion. Specifically, FCOS trains a regression network to directly predict the distance from each pixel in the object to the object's boundary. This idea is helpful for video grounding. If we train a model to predict the distance from each frame to the ground truth boundary, then all the frames within the ground truth can be leveraged as positive training samples. In this way, the number of positive samples is sufficiently increased and thus benefits the training.

In this paper, we propose a dense regression network for video grounding, which consists of four modules, including a video-query interaction module, a location regression head, a semantic matching head, and an IoU regression head. The main idea is as straightforward as training a regression module to directly regress the ground truth boundary from each frame within the ground truth. In the training, all frames within the ground truth are selected as positive samples. By doing so, the sparse annotation is able to be used to generate more positive training samples sufficiently, which boosts grounding performance eventually.

For each video-query pair, our model produces dense predictions (*i.e*., one predicted temporal bounding box for each frame) while we are only interested in the one that matches the query best. To select the best grounding result, we focus on two perspectives: 1) Does the box match the query semantically? 2) Does the box match the temporal boundary of the ground truth? Specifically, we train a semantic matching head to predict a score for each box, which indicates whether the content in the box matches the query semantically. However, this score cannot directly reflect the localization quality (*i.e*., the IoU with the ground truth), which is of vital importance for video grounding. This motivates us to further consider the localization quality of each prediction. To do so, one may use the "centerness" assumption in FCOS, which, however, is empirically found inapplicable for video grounding (see Table 5). In this paper, we train an IoU regression head to directly estimate the IoU between the predicted box and the ground truth. Last, we combine the matching score and the IoU score to find the best grounding result. It is worth noting that the dense regression network works in a one-stage manner. We evaluate our proposed method on three popular benchmarks for video grounding, *i.e*., Charades-STA [9], ActivityNet-Captions [25] and TACoS [32].

To sum up, our contributions are as follows:

- We propose a dense regression network for one-stage video grounding. We provide a new perspective to leverage dense supervision from the sparse annotations in video grounding.
- To explicitly consider the localization quality of the predictions, we propose a simple but effective IoU regression head and integrate it into our one-stage paradigm.
- We verified the effectiveness of our proposed method on three video grounding datasets. On ActivityNet-Captions especially, our method obtains the accuracy of 42.49%, which significantly outperforms the state-of-the-art, *i.e*., 36.90% by He *et al*. [16].

2. Related work

Video grounding. Recently, great progress has been achieved in deep learning [48, 47, 2, 15, 13, 14, 19, 1, 51], which facilitates the development of video grounding. Existing methods on this task can be grouped into two categories (*i.e*., two-stage and one-stage). Most two-stage methods [9, 17, 10, 4, 30, 49] resort to a propose-and-rank pipeline, where they first generate proposals and then rank them relying on the similarity between proposal and query. Gao *et al*. [9] and Hendricks *et al*. [17] 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, Xu *et al*. [40] incorporate a query into a neural network to generate the query-guided proposals. Zhang *et al*. [46] explicitly model temporal relations among proposals using a graph. The two-stage methods are straightforward but have two limitations: 1) Comparing all the proposal-query pairs leads to a heavy computation cost; 2) The performance highly relies on the quality of proposals. Our method is able to avoid the above limitations since the candidate proposals are not required.

To perform video grounding more efficiently, many methods that go beyond the propose-and-rank pipeline have been studied. He *et al*. [16] and Wang *et al*. [39] propose a reinforcement learning method for video grounding task. In the work by He *et al*. [16], the agent adjusts the boundary of a temporal sliding window according to the learned policy. At the same time, Yuan *et al*. [43] propose the attention-based grounding approach which directly predicts the temporal coordinates of the video segment that described by the input query. Ghosh *et al*. [11] and Chen *et al*. [4] propose to select the starting and ending frames by leveraging cross-modal interactions between text and video. Specifically, they predict two probabilities at each frame, which indicate whether this frame is a starting (or ending) frame of the ground truth video segment. Unlike the previous work by Chen *et al*. [4] and Ghosh *et al*. [11] where only the starting and ending frame are selected as positive training samples, our method is able to leverage much more positive training samples, which significantly boosts the grounding performance.

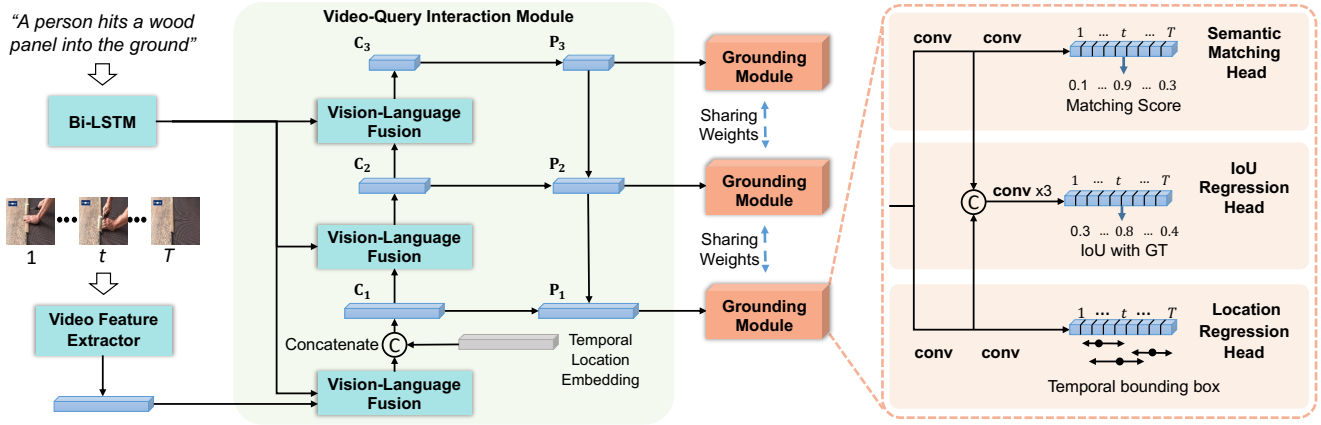

Figure 2. Schematic of our dense regression network. We use the video-query interaction module to fuse the features from the video and query. By constructing the feature pyramid, we obtain hierarchical feature maps and forward them to the grounding module. At each location t, the grounding module predicts a temporal bounding box, along with a semantic matching score and an IoU score for ranking.

Anchor-free object detection. Anchor-free object detectors [21, 31, 36, 26, 24] predict bounding boxes and class scores without using predefined anchor boxes. Redmon *et al*. propose YOLOv1 [31] to predict bounding boxes at the points near the center of objects. Law *et al*. propose CornerNet [26] to detect an object bounding box as a pair of corners and CornerNet obtains a high recall. Kong *et al*. propose FoveaBox [24] to predict category-sensitive semantic maps for the object existing possibility and produce a category-agnostic bounding box at each position. Tian *et al*. devise FCOS [36] to make full use of the pixels in a ground truth bounding box to train the model and propose centerness to suppress the low-quality predictions. Our work is related to FCOS since we also directly predict the distance from each frame to the ground truth boundary.

3. Proposed method

Notation. Let $V = \{I_t \in \mathbb{R}^{H\times W\times 3}\}_{t=1}^{T}$ be an untrimmed video, where I_t denotes the frame at time slot t with height H and width W. We denote the query with N words as $Q = \{w_n\}_{n=1}^{N}$, where w_n is the n-th word in the query.

Problem Definition. Given a video V and a query Q, video grounding requires the machine to localize a video segment (*i.e*., a temporal bounding box $\mathbf{b} = (t_s, t_e)$) starting at t_s and ending at t_e, which corresponds to the query. This task is very challenging since it is difficult to localize actions of interest precisely in a video with complex contents. More critically, only a few frames are annotated in one video, making the training samples extremely imbalanced.

3.1. General scheme

We focus on solving the problem that existing video grounding methods neglect the rich information from the frames within the ground truth, which, however, is able to significantly improve the localization accuracy. To this end, we propose a dense regression network to regress the starting (or ending) frame of the video segment described by the query for each frame. In this way, we are able to select every frame within the ground-truth as a positive training sample, which significantly benefits the training of our video grounding model.

Formally, we forward the video frames $\{I_t\}_{t=1}^{T}$ and the query $\{w_n\}_{n=1}^{N}$ to the video-query interaction module G for extracting the multi-scale feature maps. Then, each feature map is processed by the grounding module, which consists of three components, *i.e*., location regression head M_{loc}, semantic matching head M_{match} and IoU regression head M_{iou}. The **location regression head** predicts a temporal bounding box $\hat{\mathbf{b}}_t$ at the t-th frame by computing

$$\begin{aligned} \{\hat{\mathbf{b}}_t\}_{t=1}^{T} &= \{(t - \hat{d}_{t,s}, t + \hat{d}_{t,e})\}_{t=1}^{T}, \\ \{(\hat{d}_{t,s}, \hat{d}_{t,e})\}_{t=1}^{T} &= M_{loc}(G(\{I_t\}_{t=1}^{T}, \{w_n\}_{n=1}^{N})), \end{aligned} \tag{1}$$

where $(\hat{d}_{t,s}, \hat{d}_{t,e})$ are the predicted distances to the starting and ending frame. With the predicted boxes $\{\hat{\mathbf{b}}_t\}_{t=1}^{T}$ at hand, our target is to select the box that matches the query best. To this end, we propose two heads in the grounding module. The **semantic matching head** predicts a score $\hat{m}_t$ indicating whether the content in the box $\hat{\mathbf{b}}_t$ matches the query semantically. However, this score cannot directly reflect the localization quality (*i.e*., the IoU with the ground truth), which, however, is also very important for video grounding. Therefore, we propose the **IoU regression head** to predict a score $\hat{u}_t$ for directly estimating the IoU between $\hat{\mathbf{b}}_t$ and the corresponding ground truth. The schematic of our approach is shown in Figure 2. For simplicity, we denote our model as **d**ense **r**egression **n**etwork (DRN).

Inference details. Given an input video, we forward it through the network and obtain a box $\hat{\mathbf{b}}_t$, a semantic matching score $\hat{m}_t$ as well as an IoU score $\hat{u}_t$ for each frame I_t.

The final grounding result is obtained by choosing the box with the highest $\hat{m}_t \times \hat{u}_t$.

In the following, we will introduce the details of the video-query interaction module in Section 3.2. Then, we detail the location regression head, the semantic matching head and the IoU regression head in Sections 3.3, 3.4, and 3.5, respectively. Last, we introduce the training details of our model in Section 3.6.

3.2. Multi-level video-query interaction module

Building connections between vision and language is a crucial step for video grounding. To learn better vision-language representations, we propose a multi-level video-query interaction module. Given a video with T frames, we use some feature extractor (e.g., C3D [37]) to obtain the video feature $\mathbf{F} \in \mathbb{R}^{T \times c}$, where c is the channel dimension. Then, the vision-language representations are produced by using multi-level fusion and temporal location embedding.

Multi-level fusion. The target video segments described by the query often have large scale variance in video grounding. For example on Charades-STA dataset [9], the shortest ground truth is 2.4s while the longest is 180.8s. To handle this issue, we follow Lin *et al.* [27] to obtain a set of hierarchical feature maps from multiple levels. Since the model may focus on different parts of the input query at each level, we follow [18] to fuse the query and the video features at different levels. Specifically, we encode the query $Q = \{w_n\}_{n=1}^{N}$ into $\{\boldsymbol{h}_n\}_{n=1}^{N}$ and a global representation g by using a bi-directional LSTM as:

$$\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N = \text{BiLSTM}(Q) \text{ and } g = [\boldsymbol{h}_1; \boldsymbol{h}_N], \quad (2)$$

where $\boldsymbol{h}_n = [\overrightarrow{h}_n; \overleftarrow{h}_n]$ is the concatenation of the forward and backward hidden states of the LSTM for the n-th word. For the i-th level, a textual attention $\alpha_{i,n}$ is computed over the words, and the query feature $\boldsymbol{q}_i$ is computed as:

$$\begin{aligned} \boldsymbol{q}_i &= \sum_{n=1}^{N} \alpha_{i,n} \cdot \boldsymbol{h}_n, \\ \alpha_{i,n} &= \text{Softmax}(W_1(\boldsymbol{h}_n \odot (W_2^{(i)} \text{ReLU}(W_3 g)))), \end{aligned} \quad (3)$$

where $\odot$ is element-wise multiplication. W_1 and W_3 are the parameters shared across different levels but $W_2^{(i)}$ is learned separately for each level i. Given the input visual feature $\mathbf{M}_i \in \mathbb{R}^{T_i \times c}$ of a vision-language fusion module, we first duplicate $\boldsymbol{q}_i$ for T_i times to obtain a feature map $\mathbf{D}_i \in \mathbb{R}^{T_i \times c}$, where T_i is the temporal resolution at the i-th level. Then, we perform element-wise multiplication to fuse $\mathbf{M}_i$ and $\mathbf{D}_i$, leading to a set of feature maps $\{\mathbf{C}_i \in \mathbb{R}^{T_i \times c}\}_{i=1}^{L}$, where L is set to 3 in our paper. Last, we obtain the feature maps $\{\mathbf{P}_i \in \mathbb{R}^{T_i \times c}\}_{i=1}^{L}$ for the grounding module by using FPN. We put more details in the supplementary material.

Temporal location embedding. We find that the queries often contain some words for referring temporal orders, such as "after" and "before". Therefore, we seek to fuse the temporal information of the video with the visual features. The temporal location of the t-th frame (or segment) is $\boldsymbol{l}_t = [\frac{t-0.5}{T}, \frac{t+0.5}{T}, \frac{1}{T}]$. The location embedding $\boldsymbol{l}_t$ is concatenated with the output of the vision-query fusion module that fuses the video feature $\mathbf{F}$ and the query feature. Note that the concatenation is performed along the channel dimension, resulting in the feature map $\mathbf{C}_1$.

3.3. Location regression head

With the vision-language representation $\mathbf{P}$ (we omit index i for better readability), we propose a location regression head to predict the distance from each frame to the starting (or ending) frame of the video segment that corresponds to the query. We implement it as two 1D convolution layers with two output channels in the last layer. For each location t on the feature map $\mathbf{P}$, if it falls inside the ground truth, then this location is considered as a training sample. Then, we have a vector $\boldsymbol{d}_t = (d_{t,s}, d_{t,e})$ being the regression target at location t. Here, $d_{t,s}$ and $d_{t,e}$ denote the distance from location t to the corresponding boundary and are computed as

$$d_{t,s} = t - t_s, d_{t,e} = t_e - t, \quad (4)$$

where t_s and t_e is the starting and ending frames of the ground truth, respectively. It is worth noting that $d_{t,s}$ and $d_{t,e}$ are all positive real values since the positive location t falls in the ground truth (*i.e.*, $t_s < t < t_e$). For those locations fall outside the ground truth, we do not use them to train the location regression head as in [36].

It is worth mentioning that the FPN [27] exploited in our video-query interaction module could also help the location regression head. The intuition is that all the positive locations from different feature maps can be used to train the location regression head, which further increases the number of training samples.

3.4. Semantic matching head

For each video-query pair, the location regression head predicts a temporal bounding box $\hat{\mathbf{b}}_t$ at each location t. Then, how to select the box that matches the query best is the key to perform video grounding.

Since the target of video grounding is to localize the video segments described by the query, it is straightforward to evaluate whether the content in $\hat{\mathbf{b}}_t$ matches the query semantically. To this end, we devise a semantic matching head to predict a score $\hat{m}_t$ for each predicted box $\hat{\mathbf{b}}_t$. The semantic matching head is implemented as two 1D convolution layers with one output channel in the last layer. If location t falls in the ground truth, its label is set as $m_t = 1$. For those

locations fall outside the ground truth, we consider them as negative training samples, *i.e.*, $m_t = 0$.

3.5. IoU regression head

The semantic matching score $\hat{m}_t$ indicates whether the content in the box $\hat{\mathbf{b}}_t$ matches the query semantically. However, we also care about whether $\hat{\mathbf{b}}_t$ matches the ground truth temporal boundary, which can be measured by the localization quality (*i.e.*, the IoU with the ground truth).

To find the box with the best localization quality, one may use the "centerness" technique in FCOS [36]. In short, "centerness" is introduced for object detection to suppress the low-quality detected objects based on a hand-crafted assumption—the location closer to the center of objects will predict a box with higher localization quality (*i.e.*, a larger IoU with the ground truth). However, we empirically found that this assumption is inapplicable to video grounding. Specifically, we conduct an experiment to find out which location predicts the best box (*i.e.*, has the largest IoU with the ground truth). For each video-query pair, we select the predicted box that has the largest IoU with the ground truth. Then, we divide the ground truth into three portions evenly and sum up the number of locations that predicts the best box for each portion. Experimental results show that More than 46% of the predictions are not predicted by the central locations of the ground truth.

In this paper, we propose to explicitly consider the localization quality of the predicted box $\hat{\mathbf{b}}_t$ in the training and testing. The main idea is as straightforward as predicting a score at each location t to estimate the IoU between $\hat{\mathbf{b}}_t$ and the corresponding ground truth. To do so, we train a three-layer convolution network as the IoU regression head in the grounding module, as shown in Figure 2. Note that the input of the IoU regression head is the concatenation of the feature maps obtained from the first convolution layer of the semantic matching head and the location regression head. The training target u_t is obtained by calculating the IoU between $\hat{\mathbf{b}}_t$ and the corresponding ground truth.

3.6. Training details

We define the training loss function for the location regression head as follows:

$$L_{loc} = \frac{1}{N_{pos}} \sum_{t=1}^{T} \mathbb{1}_{gt}^{t} L_1(\boldsymbol{d}_t, \hat{\boldsymbol{d}}_t), \tag{5}$$

where we use the IoU regression loss [41] as L_1 following Tian *et al*. [36]. N_{pos} is the number of positive samples. $\mathbb{1}_{gt}^{t}$ is the indicator function, being 1 if location t falls in the ground truth and 0 otherwise. The training loss function for the semantic matching head is defined as:

$$L_{match} = \frac{1}{N_{pos}} \sum_{t=1}^{T} L_2(m_t, \hat{m}_t), \tag{6}$$

where we adopt the focal loss [28] as L_2 since it is effective when handling the class imbalance issue. To train the IoU regression head for predicting the IoU between the predicted box and ground truth, we define the training loss function as follows:

$$L_{iou} = \sum_{t=1}^{T} L_3(u_t, \hat{u}_t), \tag{7}$$

where we choose to use the Smooth-L1 loss [12] as L_3 because it is less sensitive to outliers.

With randomly initialized parameters, the location regression head often fails to produce high-quality temporal bounding boxes for training the IoU regression head. Thus, we propose a three-step strategy to train the proposed DRN, which consists of a video-query interaction module G, a semantic matching head M_{match}, an IoU regression head M_{iou} and a location regression head M_{loc}. Specifically, in the first step, we fix the parameters of the IoU regression head and train the DRN by minimizing Equations (5) and (6). In the second step, we fix the parameters in DRN except for the IoU regression head and train the DRN by minimizing Equation (7). In the third step, we fine-tune the whole model in an end-to-end manner[1].

4. Experiments

4.1. Datasets

Charades-STA is a benchmark dataset for the video grounding task, which is built upon the Charades [33] dataset. The Charades dataset is collected for video action recognition and video captioning. Gao *et al*. [9] adapt the Charades dataset to the video grounding task by collecting the query annotations. The Charades-STA dataset contains 6672 videos and involves 16128 video-query pairs, where 12408 pairs are used for training and 3720 for testing. The duration of the videos is 29.76 seconds on average. Each video has 2.4 annotated moments and the duration of each moment is 8.2 seconds. We follow the same split of the dataset as in Gao *et al*. [9] for fair comparisons.

ActivityNet-Captions (ANet-Captions) is collected for the dense video captioning task. It is also a popular benchmark for video grounding since the captions can be used as queries. ANet-Captions consists of 20K videos with 100K queries. The videos are associated with 200 activity classes, where the content is more diverse compared to Charades-STA. On average, each video contains 3.65 queries, and each query has an average length of 13.48 words. The average duration of the videos is around 2 minutes. The ActivityNet Captions dataset is split into the training set, validation set, testing set with a 2:1:1 ratio, including 37421, 17505 and 17031 video-query pairs separately. The public

[1] We put the training algorithm in the supplementary material.

split of the dataset contains a training set and two validation sets val_1 and val_2. The testing set is withheld for competition. We train our model on the training set and evaluate it on val_1 and val_2 separately for fair comparisons.

TACoS dataset is collected by Regneri *et al*. [32] for video grounding and dense video captioning tasks. It consists of 127 videos on cooking activities with an average length of 4.79 minutes. For the video grounding task, TACoS dataset contains 18818 video-query pairs. Compared to ActivityNet Captions dataset, TACoS has more temporally annotated video segments with queries per video. Each video has 148 queries on average. Moreover, TACoS dataset is very challenging since the queries in TACoS dataset span over only a few seconds even a few frames. We follow the same split of the dataset as Gao *et al*. [9] for fair comparisons, which has 10146, 4589, and 4083 video-query pairs for training, validation, and testing respectively.

4.2. Implementation details

Evaluation metric. For fair comparisons, we follow Gao *et al*. [9] to compute "R@n, IoU=m" as the evaluation metric. To be specific, it represents the percentage of testing samples that have at least one correct grounding prediction (*i.e*., the IoU between the prediction and the ground truth is larger than m) in the top-n predictions.

Video Feature Extractor. We use the C3D [37] network pre-trained on Sports-1M [22] as the feature extractor. The C3D network takes 16 frames as input and the outputs of the *fc6* layer with dimensions of 4096 are used as a feature vector. We also extract the I3D [3] and VGG [34] features to conduct experiments on Charades-STA. More details about the feature extractor are put in the supplementary material.

Language Feature. We transform each word of language sentences into lowercase. We use pre-trained GloVe word vectors to initialize word embeddings with the dimension of 300. A one-layer bi-directional LSTM with 512 hidden units serves as the query encoder.

Training settings. The learning rate in the first training step is 0.001 and we decay it by a factor of 100 for the second step. During fine-tuning, we set the learning rate as 10^{-6}. We set batch size as 32 and use Adam [23] as the optimizer.

4.3. Comparisons with state-of-the-arts

Comparisons on Charades-STA. We compare our model with the state-of-the-art methods in Table 1. Our DRN reaches the highest scores over all IoU thresholds. Particularly, when using the same C3D features, our DRN outperforms the previously best method (*i.e*., R-W-M [16]) by 8.7% absolute improvement, in terms of R@1, IoU=0.5. For fair comparisons with MAN [46] and ExCL [11], we perform additional experiments by using the same features (*i.e*., VGG and I3D) as reported in their papers. Our DRN outperforms them by 1.66% and 8.99%, respectively.

Table 1. Comparisons with state-of-the-arts on Charades-STA.

Methods	Feature	R@1 IoU=0.5	R@1 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.7
CTRL [9]	C3D	23.63	8.89	58.92	29.52
SMRL [39]	C3D	24.36	11.17	61.25	32.08
MAC [10]	C3D	30.48	12.20	64.84	35.13
T-to-C [40]	C3D	35.60	15.80	79.40	45.40
R-W-M [16]	C3D	36.70	-	-	-
DRN (ours)	C3D	**45.40**	**26.40**	**88.01**	**55.38**
ExCL [11]	I3D	44.10	22.40	-	-
DRN (ours)	I3D	**53.09**	**31.75**	**89.06**	**60.05**
SAP [7]	VGG	27.42	13.36	66.37	38.15
MAN [46]	VGG	41.24	20.54	83.21	51.85
DRN (ours)	VGG	**42.90**	**23.68**	**87.80**	**54.87**

Table 2. Comparisons on ANet-Captions using C3D features. (*) indicates the method that uses val_2 split as the testing set, while other methods use the val_1 split.

Methods	R@1 IoU=0.5	R@1 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.7
CTRL [9]	14.00	-	-	-
ACRN [29]	16.17	-	-	-
T-to-C[40]	27.70	13.60	59.20	38.30
R-W-M [16]	36.90	-	-	-
DRN (ours)	**42.49**	**22.25**	**71.85**	**45.96**
TGN* [4]	27.93	-	44.20	-
ABLR* [43]	36.79	-	-	-
CMIN* [49]	43.40	23.88	67.95	**50.73**
DRN* (ours)	**45.45**	**24.36**	**77.97**	50.30

Table 3. Comparisons on TACoS using C3D features.

Methods	R@1 IoU=0.5	R@5 IoU=0.5
ABLR [43]	9.40	-
CTRL [9]	13.30	25.42
ACRN [29]	14.62	24.88
SMRL [39]	15.95	27.84
CMIN [49]	18.05	27.02
TGN [4]	18.90	31.02
MAC [10]	20.01	30.66
DRN (ours)	**23.17**	**33.36**

Comparisons on ActivityNet-Captions. Table 2 reports the video grounding results of various methods. We follow the previous methods to use C3D features for fair comparisons. Since previous methods use different testing splits, we report the performance of our model on both val_1 and val_2. Regarding R@1, IoU=0.5, our method outperforms R-W-M [16] by 5.59% absolute improvement on val_1 split and exceeds CMIN [49] by 2.05% on val_2 split.

Comparisons on TACoS. We compare our DRN with state-of-the-art methods with the same C3D features in Table 3. It is worth noting that this dataset is very challenging since each video may correspond to multiple queries (148 queries

Table 4. Ablation study on the number of positive training samples on Charades, measured by R@1 and R@5 when IoU=0.5.

Methods	R@1 IoU=0.5	Gain	R@5 IoU=0.5	Gain
DRN-Center	38.36	-	83.36	-
DRN-Random	40.88	2.52	84.11	0.75
DRN-Half	42.79	4.43	85.88	2.52
DRN-All	**45.40**	**7.04**	**88.01**	**4.65**

on average). Despite its difficulty, our method reaches the highest score in terms of both R@1 and R@5 when $IoU = 0.5$ and outperforms previous best result by a large margin (*i.e.*, 23.17% vs. 20.01%).

5. Ablation studies

In this section, we will perform complete and in-depth ablation studies to evaluate the effect of each component of our model. More details about the structures and training configurations of the baseline methods (such as DRN-Center) can be found in the supplementary material.

5.1. How does location regression help?

Compared with other one-stage video grounding methods, the key to our DRN is to leverage more positive samples for the training. Here, we implement three variants of our methods: **DRN-Half**, **DRN-Random** and **DRN-Center**. The three baselines are the same as the original DRN (**DRN-All**) except that they only select a subset of frames within the ground truth as the positive training samples. Specifically, DRN-Half randomly chooses 50% of the frames within the ground truth to train the model. DRN-Random and DRN-Center are the extreme cases of our location regression settings, where they only randomly select one frame or the center frame within the ground truth as the positive training sample. By comparing the performance of the variants with our DRN, we justify the importance of increasing the number of positive training samples to train a one-stage video grounding model. Table 4 shows that all of these variants decrease the performance significantly. It verifies the effectiveness of our dense regression network, which is able to mine more positive training samples from sparse annotations.

5.2. Does IoU regression help video grounding?

As discussed in Section 3.5, besides the IoU regression head, using "centerness" technique is another way to assess the localization quality. Here, we implement a variant of our model by replacing the IoU regression head with the centerness head in FCOS [36]. Specifically, the centerness head is trained to predict a centerness score at each frame. The frame closer to the ground truth's center is expected to have a larger centerness value. In the inference, we follow [36] to multiply the centerness score and matching score to obtain the final score for each predicted box. We also implement a baseline by removing the IoU regression head from our model and directly use the matching score to rank the predictions. Table 5 reveals that the IoU regression head consistently improves the performance on both datasets. These results demonstrate that the matching score is not sufficient to evaluate the localization quality. Predicting the IoU between the predicted box and ground truth is straightforward and helpful for video grounding. Using centerness slightly decreases the grounding accuracy since the centerness assumption is not suitable for video grounding. We also visualize the qualitative results in Figure 3. In the top example, the two grounding results are both predicted by the frames within the ground truth, while the IoU regression head helps to select the one that has a larger IoU. In the bottom example, the background context is similar and the query is complex. Despite such difficulty, the IoU regression head still helps to select a better grounding result. More visualization results are shown in the supplementary material.

Table 5. Ablation study of the IoU regression head on Charades-STA and ActivityNet-Captions, measured by R@1 when IoU=0.5.

Dataset	Methods	R@1 IoU=0.5
Charades-STA	w/o IoU regression head	44.13
	w/ Centerness	44.02
	w/ IoU regression head	**45.40**
ANet-Captions	w/o IoU regression head	40.44
	w/ Centerness	39.83
	w/ IoU regression head	**42.49**

Table 6. Ablation study of multi-level fusion (MLF) and location embedding on Charades-STA, measured by R@1 when IoU=0.5.

Dataset	Components		R@1
	MLF	location	IoU=0.5
Charades-STA	×	×	43.04
	✓	×	43.79
	×	✓	43.47
	✓	✓	**45.40**
ANet-Captions	×	×	39.78
	✓	×	40.61
	×	✓	40.96
	✓	✓	**42.49**

5.3. Does multi-level fusion help?

The multi-level fusion (MLF) technique extracts different representations of the same query at different levels and fuses them with the video feature. Here, we implement a baseline by removing MLF from our DRN. Specifically, we only fuse the visual feature and the query feature at the first level (*i.e.*, C_1 in Figure 2). From Table 6, applying MLF to our model is able to lift the video grounding performance

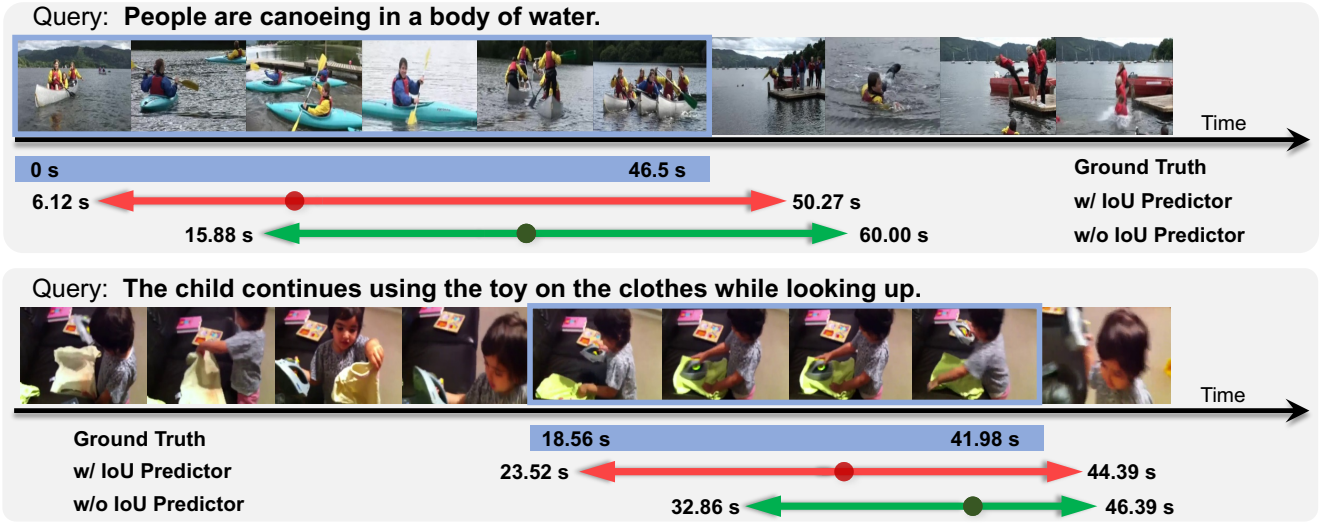

Figure 3. Qualitative results on ActivityNet Captions dataset.

Table 7. Ablation study of the location embedding on the collected subset of ANet-Captions, measured by R@1 when IoU=0.5.

Train	Test	Methods	R@1 IoU=0.5
Full-set	Full-set	w/o location	40.61
		w/ location	**42.49**
Full-set	Sub-set	w/o location	47.38
		w/ location	**48.37**
Sub-set	Sub-set	w/o location	43.28
		w/ location	**44.97**

on both Charades-STA (43.79% vs. 43.04%) and ANet-Captions datasets (40.61% vs. 39.78%). In addition, we implement another baseline **MLF-Same** by using the same query feature to fuse the video feature at different levels. In our experiments, the **MLF-Same** baseline performs worse than our DRN on Charades-STA (44.76% vs. 45.40%), revealing that extracting different query features at different levels is able to improve the video-query representations and boost the grounding performance eventually.

5.4. How does the location embedding help?

To evaluate the effectiveness of the temporal location embedding in our model, we conduct an ablation study by directly forwarding the video features into the network without concatenating with the location embeddings. The results in Table 6 conclude that the location embedding makes the localization more precisely. One possible reason is that the model is able to learn the temporal orders of the video contents through the location embeddings. To further study the effect of the location embedding, we collect a "temporal" subset of samples from the ANet-Captions dataset. In particular, we are interested in the query that contains four commonly used temporal words (*i.e.*, before, after, while, then). The subset consists of 7176 training samples and 3620 testing samples. We use two settings to evaluate our model: 1) train on full ANet-Captions dataset and test on the temporal subset; 2) train and test on the temporal subset. From Table 7, using location embedding consistently improves the performance in both settings. Especially when training and testing the model on the temporal subset, the performance gain increases to 1.7%, further verifying the effectiveness of the location embedding.

6. Conclusions

In this paper, we have proposed a dense regression network for video grounding. By training the model to predict the distance from each frame to the starting (ending) frame of the video segment described by the query, the number of positive training samples is significantly increased, which boosts the performance of video grounding. Moreover, we have devised a simple but effective IoU regression head to explicitly consider the quality of localization results for video grounding. Our DRN outperforms the state-of-the-art methods on three benchmarks, *i.e.*, Charades-STA, ActivithNet-Captions and TACoS. It would be interesting to extend our DRN for temporal action localization and dense video captioning, and we leave it for our future work.

Acknowledgements. This work was partially supported by Guangdong Provincial Scientific and Technological Funds under Grant 2018B010107001, Grant 2019B010155002, National Natural Science Foundation of China (NSFC) 61836003 (key project), Program for Guangdong Introducing Innovative and Enterpreneurial Teams 2017ZT07X183, Tencent AI Lab Rhino-Bird Focused Research Program (No. JR201902), Fundamental Research Funds for the Central Universities D2191240.

References

[1] Jiezhang Cao, Yong Guo, Qingyao Wu, Chunhua Shen, Junzhou Huang, and Mingkui Tan. Adversarial learning with local coordinate coding. In *International Conference on Machine Learning*, 2018. 2

[2] Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, and Mingkui Tan. Multi-marginal wasserstein gan. In *Advances in Neural Information Processing Systems*, pages 1774–1784, 2019. 2

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017. 1, 6

[4] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 162–171, 2018. 2, 6

[5] Jingyuan Chen, Lin Ma, Xinpeng Chen, Zequn Jie, and Jiebo Luo. Localizing natural language in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8175–8182, 2019. 1

[6] Peihao Chen, Chuang Gan, Guangyao Shen, Wenbing Huang, Runhao Zeng, and Mingkui Tan. Relation attention for temporal action localization. *IEEE Transactions on Multimedia*, 2019. 1

[7] Shaoxiang Chen and Yu-Gang Jiang. Semantic proposal for activity localization in videos via sentence query. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8199–8206, 2019. 6

[8] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9346–9355, 2019. 1

[9] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5267–5275, 2017. 1, 2, 4, 5, 6

[10] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. Mac: Mining activity concepts for language-based temporal localization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 245–253. IEEE, 2019. 1, 2, 6

[11] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann. Excl: Extractive clip localization using natural language descriptions. In *The North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 1, 2, 6

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 5

[13] Yong Guo, Jian Chen, Qing Du, Anton Van Den Hengel, Qinfeng Shi, and Mingkui Tan. Multi-way backpropagation for training compact deep neural networks. *Neural networks*, 2020. 2

[14] Yong Guo, Qi Chen, Jian Chen, Qingyao Wu, Qinfeng Shi, and Mingkui Tan. Auto-embedding generative adversarial networks for high resolution image synthesis. *TMM*, 2019. 2

[15] Yong Guo, Yin Zheng, Mingkui Tan, Qi Chen, Jian Chen, Peilin Zhao, and Junzhou Huang. Nat: Neural architecture transformer for accurate and compact architectures. In *Advances in Neural Information Processing Systems*, pages 735–747, 2019. 2

[16] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8393–8400, 2019. 2, 6

[17] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1380–1390, 2018. 2

[18] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2018. 4

[19] Zhibin Hu, Yongsheng Luo, Jiong Lin, Yan Yan, and Jian Chen. Multi-level visual-semantic alignments with relation-wise dual attention network for image and text matching. 2

[20] Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. Location-aware graph convolutional networks for video question answering. In *The AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 1

[21] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *CoRR*, abs/1509.04874, 2015. 2, 3

[22] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014. 6

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015. 6

[24] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Foveabox: Beyond anchor-based object detector. *CoRR*, abs/1904.03797, 2019. 2, 3

[25] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 706–715, 2017. 2

[26] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 2, 3

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017. 4

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 5

[29] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. Attentive moment retrieval in videos. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 15–24. ACM, 2018. 6

[30] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. Cross-modal moment localization in videos. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 843–851. ACM, 2018. 2

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 2, 3

[32] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics (TACL)*, 1:25–36, 2013. 2, 6

[33] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 510–526. Springer, 2016. 5

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR)*, 2015. 6

[35] Joyeeta Singha, Amarjit Roy, and Rabul Hussain Laskar. Dynamic hand gesture recognition using vision-based approach for human–computer interaction. *Neural Computing and Applications*, 29(4):1129–1141, 2018. 1

[36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 4, 5, 7

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. 4, 6

[38] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–36. Springer, 2016. 1

[39] Weining Wang, Yan Huang, and Liang Wang. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 334–343, 2019. 2, 6

[40] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 9062–9069, 2019. 1, 2, 6

[41] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520. ACM, 2016. 5

[42] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487, 2018. 1

[43] Yitian Yuan, Tao Mei, and Wenwu 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 (AAAI)*, volume 33, pages 9159–9166, 2019. 1, 2, 6

[44] Runhao Zeng, Chuang Gan, Peihao Chen, Wenbing Huang, Qingyao Wu, and Mingkui Tan. Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization. *IEEE Transactions on Image Processing*, 2019. 1

[45] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 1

[46] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry 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 (CVPR)*, pages 1247–1257, 2019. 2, 6

[47] Yifan Zhang, Hanbo Chen, Ying Wei, Peilin Zhao, Jiezhang Cao, et al. From whole slide imaging to microscopy: Deep microscopy adaptation network for histopathology cancer image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, page 360?368. Springer, 2019. 2

[48] Yifan Zhang, Ying Wei, Peilin Zhao, Shuaicheng Niu, et al. Collaborative unsupervised domain adaptation for medical image diagnosis. In *Medical Imaging meets NeurIPS*, 2019. 2

[49] Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. 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)*, pages 655–664, 2019. 2, 6

[50] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[51] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 875–886, 2018. 2

Supplementary Material
Dense Regression Network for Video Grounding

In the supplementary material, we first give the training details of our DRN in Section A. Then, we illustrate the details of the video-query interaction module in Section B. Next, we detail the grounding module in Section C, followed by more qualitative results in Section D. Last, we provide more details of "centerness" in Section E. .

A. More details about our DRN

The training details of our proposed DRN are shown in Algorithm 1. With randomly initialized parameters, the location regression head often fails to produce high-quality temporal bounding box for training the IoU regression head. Thus, we propose a three-step strategy to train the proposed DRN. Specifically, in the first step, we fix the parameters of the IoU regression head and train the DRN by minimizing Equations (5) and (6). In the second step, we fix the parameters in DRN except for the IoU regression head and train the DRN by minimizing Equation(7). In the third step, we fine-tune the whole model in an end-to-end manner.

Algorithm 1 Training details of DRN.

Input: Video $V = \{I_t\}_{t=1}^{T}$; query $Q = \{w_n\}_{n=1}^{N}$

Step1: Fix the parameters of M_{iou}

1: **while** not converges **do**
2: predict matching score $\hat{m}_t$
3: predict regression offset $\hat{\boldsymbol{d}}_t$ using Equation (1)
4: update DRN by minimizing Equations (5) and (6)
5: **end while**

Step2: Fix the parameters of G, M_{match}, and M_{loc}

1: **while** not converges **do**
2: predict bounding box $\hat{\mathbf{b}}_t$ using Equation (1)
3: predict IoU between $\hat{\mathbf{b}}_t$ and ground truth
4: update DRN by minimizing Equation (7)
5: **end while**

Step3: Fine-tune G, M_{match}, M_{loc}, and M_{iou} jointly

1: **while** not converges **do**
2: predict matching score $\hat{m}_t$
3: predict bounding box $\hat{\mathbf{b}}_t$ using Equation (1)
4: predict IoU between $\hat{\mathbf{b}}_t$ and ground truth
5: update DRN by minimizing Equations (5), (6), (7)
6: **end while**

Output: Trained DRN

B. Details of video-query interaction module

The video-query interaction module consists of two parts, as shown in Figure A. The first part serves as a data preprocessor, which takes the query sentences, video frames and temporal coordinates as input and outputs the query feature and video feature (C_1). The second part is a fully convolutional network with vision-language fusion modules. It is used to fuse the video feature and query feature and construct a feature pyramid.

B.1. Video feature extractor

Instead of predicting a temporal bounding box at each frame, we exploit a more efficient way to implement our dense regression network. Specifically, we divide a video into K segments evenly. Thus, the temporal resolution of the video comes to K, which significantly reduces the computation in our model. Then, we use our model to predict a temporal bounding box w.r.t. the central frame of each segment. We set K as 32 for Charades-STA and ActivityNet Captions, and 128 for TACoS dataset. Three types of feature extractor are detailed as follows:

C3D. We use C3D [37] pre-trained on sport1M [22] to extract features. The C3D network takes 16 consecutive frames (a snippet) as input and the output of *fc*6 layer is used as a snippet-level feature vector. The feature of each segment is obtained by performing max-pooling among the snippet-level features that correspond to the segment.

VGG. We use VGG16-BN [34] pre-trained on ImageNet. VGG16-BN takes one frame as input and the output of *fc*7 layer is used as the frame-level feature. The segment feature is obtained by performing max-pooling among the frame-level features that correspond to the segment.

I3D. We use I3D [3] pre-trained on Kinetics to extract features. The I3D network takes 64 consecutive frames (a snippet) as input and outputs a snippet-level feature vector. The feature of each segment is obtained by performing max-pooling among the snippet-level features that correspond to the segment.

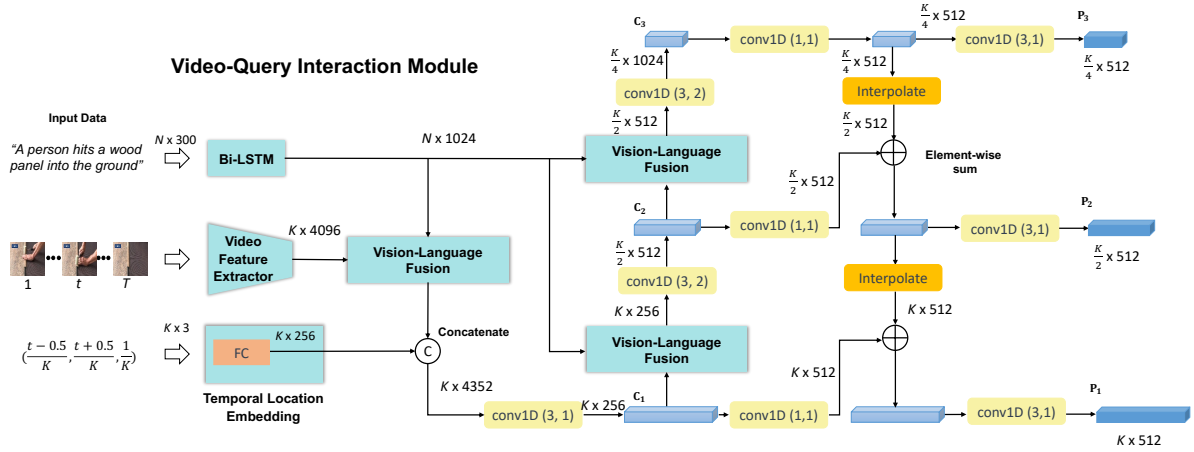

Figure A. The details of Video-Query Interaction Module. Note that "Conv1D (b, s)" denotes a 1D convolution layer with a kernel size of b and a stride of s. All the convolution layers are followed by batch normalization and ReLU.

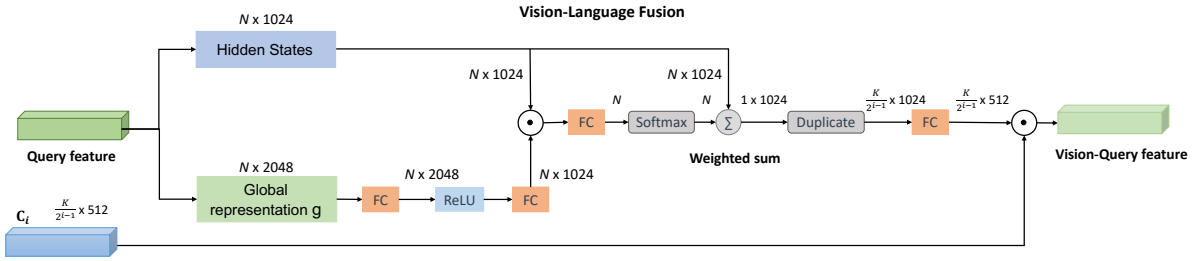

Figure B. The details of Vision-Language-Fusion Module.

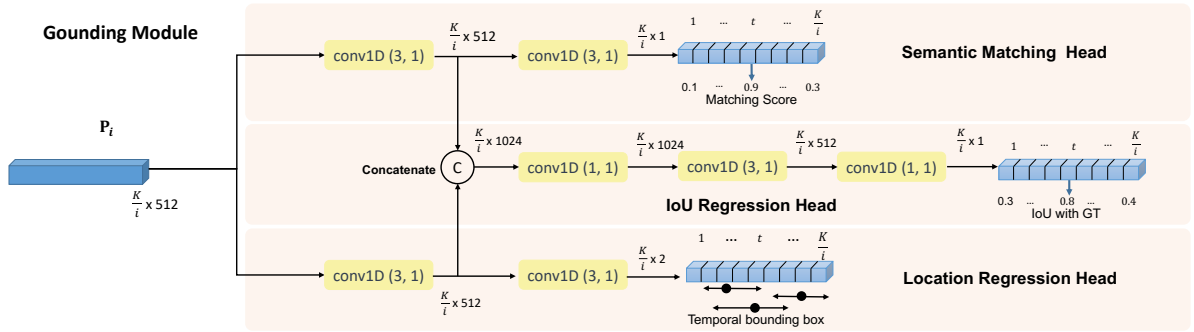

Figure C. The details of Grounding Module. The input $\mathbf{P}_i$ is from the i-th level in the feature pyramid with a temporal dimension of $\frac{K}{2^{i-1}}$.

B.2. Query feature extractor

First, each word in the input query sentence is mapped into a 300-dim vector using pre-trained GloVe word embeddings. Then, the word embeddings of the query sentence are fed into a one-layer bi-directional LSTM with 512 units. Last, the sequence of hidden states is used as query features. The hidden states of the first and the last word are concatenated, leading to the global representation g.

B.3. Location embedding

The input temporal coordinates of the k-th segment is a 3D vector, *i.e.*, $(\frac{k-0.5}{K}, \frac{k+0.5}{K}, \frac{1}{K})$. We forward it to a linear layer, leading to a 256D location embedding. The location embedding is then concatenated with the video features along the channel dimension.

B.4. Vision-Language Fusion Module

We apply the textual attention mechanism to the input query feature and obtain the attended feature. Then, the attended query features and the features from a lower level of the pyramid are fused by using element-wise multiplication. The details are shown in Figure B.

C. More details about grounding module

The grounding module involves three components, including semantic matching head, location regression head and IoU regression head. Both of the semantic matching

Figure D. Qualitative results.

head and location regression head consist of two 1D convolution layers, and IoU regression head contains three 1D convolution layers. The details are shown in Figure C.

D. More visualization examples

We show more qualitative results of the IoU regression head in Figure D. The IoU regression head helps to select the prediction that has a larger IoU with the ground truth.

E. More details about centerness

E.1. Details of centerness baseline

To compare our IoU regression with the centerness in FCOS [28], we conduct an experiment by replacing the loss function of IoU regression head with a centerness loss as in [28]. Specifically, we train the model to predict a centerness score for each location. The training target is defined as:

$$centerness* = \sqrt{\frac{min(d^*_{t,s}, d^*_{t,e})}{max(d^*_{t,s}, d^*_{t,e})}} \quad (1)$$

where $d^*_{t,s}$, $d^*_{t,e}$ are the distances between location t and the starting frame, the ending frame of ground truth boundary respectively. We follow [28] to adopt the binary cross-entropy loss as the loss function for centerness in our experiments.

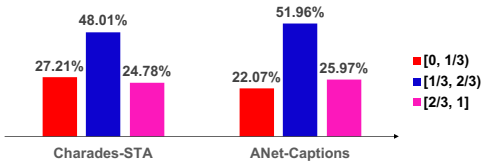

Figure E. The location distribution of the "best location" on two datasets. Here, the "best location" denotes the location that predicts the best grounding result for each video-query pair. We show the statistics of their relative locations w.r.t. the ground truth, which has been divided into three portions evenly. Here, we only focus on the locations within the ground truth since few locations fall outside of the ground truth.

E.2. Results of the centerness assumption

The centerness assumption [28] is that the location closer to the center of objects will predict a box with higher localization quality (*i.e.*, a larger IoU with the ground truth). We conduct an experiment to find out which location predicts the best box. In our experiment, we train a model using the semantic matching loss and location regression loss. For each video-query pair, we select the predicted box that has the largest IoU with the ground truth. Then, we divide the ground truth into three portions evenly and sum up the number of the locations that predicts the best box for each portion. From Figure E, more than 48% of the predictions are not predicted by the central locations of the ground truth.