Rethinking the Bottom-Up Framework for Query-based Video Localization

Long Chen,[1*] **Chujie Lu,**[1*] **Siliang Tang,**[1†] **Jun Xiao,**[1] **Dong Zhang,**[2] **Chilie Tan,**[3] **Xiaolin Li**[3,4]
[1] DCD Lab, College of Computer Science, Zhejiang University,
[2] Nanjing University of Science and Technology [3] Tongdun Technology [4] University of Florida
{longc, siliang, junx}@zju.edu.cn; chujielu@outlook.com;
dongzhang@njust.edu.cn; chilie.tan@tongdun.net; andyli@ece.ufl.edu

Abstract

In this paper, we focus on the task query-based video localization, *i.e.*, localizing a query in a long and untrimmed video. The prevailing solutions for this problem can be grouped into two categories: i) *Top-down* approach: It pre-cuts the video into a set of moment candidates, then it does classification and regression for each candidate; ii) *Bottom-up* approach: It injects the whole query content into each video frame, then it predicts the probabilities of each frame as a ground truth segment boundary (*i.e.*, start or end). Both two frameworks have respective shortcomings: the top-down models suffer from heavy computations and they are sensitive to the heuristic rules, while the performance of bottom-up models is behind the performance of top-down counterpart thus far. However, we argue that the performance of bottom-up framework is severely underestimated by current unreasonable designs, including both the backbone and head network. To this end, we design a novel bottom-up model: Graph-FPN with Dense Predictions (GDP). For the backbone, GDP firstly generates a frame feature pyramid to capture multi-level semantics, then it utilizes graph convolution to encode the plentiful scene relationships, which incidentally mitigates the semantic gaps in the multi-scale feature pyramid. For the head network, GDP regards all frames falling in the ground truth segment as the foreground, and each foreground frame regresses the unique distances from its location to bi-directional boundaries. Extensive experiments on two challenging query-based video localization tasks (natural language video localization and video relocalization), involving four challenging benchmarks (TACoS, Charades-STA, ActivityNet Captions, and ActivityVRL), have shown that GDP surpasses the state-of-the-art top-down models.

Introduction

Query-based Video Localization (QBVL), *i.e.*, capturing the gist of a query and localizing the semantic-similar moment with the query in a long and untrimmed reference video, is one of the core tasks in video scene understanding. With the release of large scale video datasets and developments in video representation learning, two challenging QBVL tasks were recently proposed: 1) **Natural Language Video Localization (NLVL)** (Gao et al. 2017; Anne Hendricks et al. 2017), where the query is a natural language description (Figure 1 (a)). 2) **Video Relocalization (VRL)** (Feng et al. 2018), where the query is a video clip (Figure 1 (b)). Both tasks share the same target, *i.e.*, identifying the start and end point of the segment in reference video which semantically corresponds to the query. Moreover, QBVL is an indispensable technique for many important video applications, *e.g.*, text-/context- based video highlight detection or retrieval, video-based person re-id (Liu et al. 2016; Ye et al. 2017; Wang et al. 2012b; 2012a; 2017).

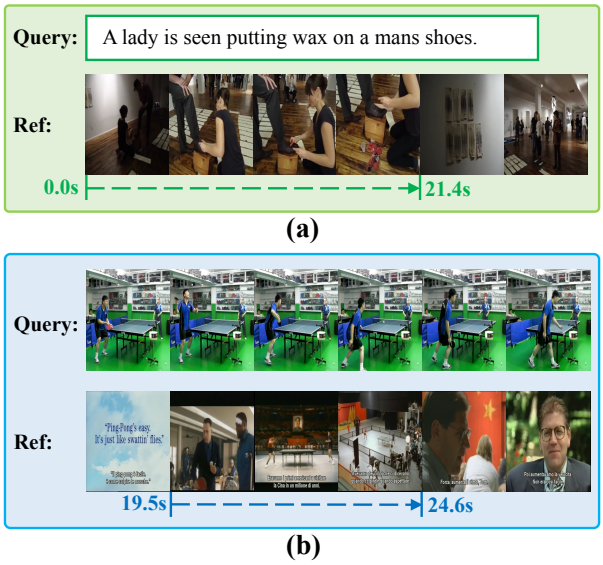

Figure 1: (a) **Natural Language Video Localization**: The query is a natural language; (b) **Video Relocalization**: The query is a video clip. Both tasks aim to localize a segment with start (0.0s/19.5s) and end (21.4s/24.6s) in the reference video (Ref) which semantically corresponds to the query.

A straightforward solution for QBVL is in a *sliding-window* fashion: it *explicitly* pre-cuts the reference video into a set of moment candidates by multiple predefined tem-

[*] L. Chen and C. Lu are co-first authors with equal contributions.
[†] Siliang Tang is the corresponding author.

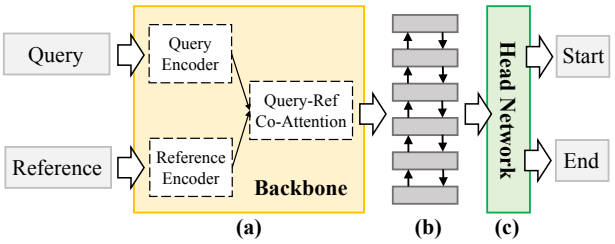

Figure 2: A typical bottom-up framework for QBVL. It always consists of a backbone (a) for query-ref interaction modeling and a head network (c) for boundaries prediction.

poral scales. After it extracts features for the query and each candidate. The QBVL degrades into a similarity matching problem (Gao et al. 2017; Anne Hendricks et al. 2017; Liu et al. 2018b; 2018c; Ge et al. 2019; Xu et al. 2019; Chen and Jiang 2019). However, these methods overlook the fruitful long term visual context in the whole video, which is helpful for deep video understanding (Wu et al. 2019). To benefit from this intuition, some QBVL models resort to RNN to encode the whole video, and *implicitly* "cut" the video by multiple temporal anchors. These temporal anchors follow the same spirits as anchor boxes in object detection (Ren et al. 2015). Finally, they do classification and regression for each candidate as the sliding-window models.

Although the ***top-down*** models (*i.e.*, sliding-window and anchor-based) have dominated the performance. It is worth noting that this framework has several notorious limitations: 1) the performance is sensitive to heuristic rules (*e.g.*, temporal scales and candidates numbers). 2) To achieve high recall, the model has to densely place candidates, which results in heavy computation and slow localization speed.

To avoid these inherent drawbacks in top-down framework, several recent models solve the QBVL in a ***bottom-up*** manner (Chen et al. 2019a; Yuan, Mei, and Zhu 2019; Feng et al. 2018). As shown in Figure 2, a bottom-up model consists of two components: a backbone (a) and a head network (c). The backbone, which is typically equipped with a co-attention or cross-gating mechanism, aims to inject the gist of the query into each reference video frame[1]. The output of the backbone is a query-ref frame sequence (b), which is always encoded by an RNN. Since the nature of the head network in bottom-up framework, *i.e.*, it predicts the probability of each frame as a boundary, the query-ref frame sequence needs to keep the same temporal resolution as the reference video. Although these models eliminate the shortcomings in top-down framework, their performance is still behind the performance of top-down models thus far, especially for long videos (*e.g.*, TACoS). We argue that the main reasons come from the current unreasonable designs:

Backbone: 1) Each video contains abundant "scene" (a cluster of frames) changes, *i.e.*, different scenes are interleaved in a video sequence. Thus, exploiting scene relationships is crucial to understand the whole video content. However, the backbone only utilizes RNN to encode frame-level interactions and ignores plentiful scene relationships. 2) To generate high-resolution query-ref frame sequence, all intermediate frame features in the backbone keep the same temporal resolution as reference video all the time. This is similar to the high-resolution feature map in ConvNet, which purely encodes low semantics (Chen et al. 2018b; Lin et al. 2017). Instead, the bottom-up framework requires each frame feature to capture higher (*i.e.*, global) semantics.

Head network: 1) To predict the probabilities of each frame as a boundary, the head network in existing bottom-up models only regards two extract boundary frames as foreground, and all other frames as background. This results in an extreme imbalance between positive and negative samples, even worse for long videos. 2) The predictions of the start and end boundaries are independent, *i.e.*, the model overlooks the content consistency between two predictions.

In this paper, we propose a novel bottom-up model: **G**raph-FPN with **D**ense **P**redictions (GDP), which mitigates all the above mentioned problems in the existing bottom-up framework. **For backbone**, GDP introduces a Graph-FPN layer to enhance the output of backbone. Specifically, it firstly constructs a pyramid hierarchy from the query-ref frame sequence (Figure 2 (b)), which helps to capture higher semantics. Then it maps all these multi-scale frame features to a scene space, where each node represents a scene. And it conducts graph convolution over all nodes in the scene space. The graph convolution not only exploits the plentiful scene relationships but also mitigates the semantic gaps between multi-scale features. Finally, these scene nodes are projected to compose new frame features. **For head network**, GDP replaces the sparse boundary predictions with dense predictions. It regards all frames falling in the ground truth segment as foreground. Each foreground frame regresses the unique distances from its location to bi-directional boundaries. Meanwhile, each frame predicts a confidence score to rank its boundaries prediction. In this manner, we utilize as many positive samples as possible to alleviate the imbalanced problem. Meanwhile, since two boundary predictions are based on a same frame feature, *i.e.*, two predictions act as a whole, which helps to avoid falling into the local optimum caused by independent predictions.

We demonstrate the effectiveness of GDP on two challenging QBVL tasks: natural language video localization over TACoS (Regneri et al. 2013), Charades-STA (Gao et al. 2017), ActivityNet Captions (Krishna et al. 2017) and video relocalization over ActivityNet-VRL (Feng et al. 2018). Without bells and whistles, GDP achieves a new state-of-the-art performance over all benchmarks and metrics.

Related Work

Query-based Video Localization

Natural Language Video Localization (NLVL). NLVL is a difficult QBVL task which involves two different modalities. The current NLVL models, which are mainly top-down models, focus on designing stronger multi-modal interaction backbone. The backbone typically contains an attention mechanism (Chen et al. 2017), *e.g.*, video-based query at-

[1]The frame in this paper is a general description for a frame in a video sequence or an element in a video frame feature sequence.

tention (Liu et al. 2018b), query-based video attention (Liu et al. 2018c), or query-video co-attention (Chen et al. 2018a; 2019a; Yuan, Mei, and Zhu 2019). To the best of our knowledge, there are two exceptions, which are neither top-down nor bottom-up models: RWM (He et al. 2019) and SM-RL (Wang, Huang, and Wang 2019). They formulate NLVL as a sequence decision making problem, solved by policy gradient (Chen et al. 2019b; Liu et al. 2018a). The action space is temporal box transformation or frame hopping.

Video Relocalization (VRL). VRL is a recently proposed QBVL task. The main challenges for VRL come from the huge differences between the query and reference video even though they express the same visual concept, *e.g.*, the appearance of environments, objects, and viewpoints. The state-of-the-art VRL method is the bottom-up model: CGBM (Feng et al. 2018). It contains a cross-gating bilinear matching in backbone to encode query-reference interaction, and a sparse head network to predict boundaries.

Top-Down vs. Bottom-Up

The concepts about the top-down and bottom-up in QBVL are similar to the one in object detection. After the appearance of anchor boxes in modern object detectors (Ren et al. 2015), top-down models have dominated object detection for years. Recently, some works start to borrow ideas from keypoint estimation and directly predict the key points of object bounding boxes (Law and Deng 2018; Zhou, Zhuo, and Krahenbuhl 2019; Zhou, Wang, and Krahenbuhl 2019; Tian et al. 2019). These bottom-up models not only enjoy much faster detection speed but also get comparable performance with the top-down models. Thus, the bottom-up detectors begin to gain unprecedented attention, which encourages us to design a stronger bottom-up QBVL model.

Graph-based Global Reasoning

Modeling context, especially for the global context, is a crucial step in many computer vision tasks. Graph-based global reasoning is a recent proposed global context modeling technique for visual recognition, which performs higher-level reasoning over a graph structure (Chen et al. 2019c; Li and Gupta 2018; Liang et al. 2018; Zhang, Yan, and He 2019). Specifically, it projects visual features in the coordinate space into a graph space, and each node in the graph updates its feature by graph convolution. Then these nodes are mapped back to the coordinate space. Different from the existing works which only consider single scale features, GDP does graph convolution over nodes from multi-scales.

Approach

The QBVL task considered in this paper, is defined as follows. Given an untrimmed reference video $\mathcal{V}$ and a query $\mathcal{Q}$ (*e.g.*, natural language or video clip), QBVL needs to predict two time points (t_s, t_e), where the segment in $\mathcal{V}$ from time point t_s to t_e corresponds to the same semantic as $\mathcal{Q}$.

In this section, we firstly introduce the architecture details about each component of GDP (Figure 3), including a query-ref interaction backbone (a), a Graph-FPN layer (b), and a dense head network (c). Then we demonstrate the details about the training and test stage of the GDP.

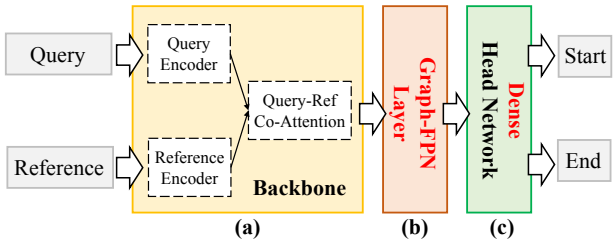

Figure 3: The architecture of GDP. It consists of a backbone (a), a Graph-FPN layer (b), and a dense head network (c).

Backbone

As shown in Figure 3, the backbone has two inputs: query feature $\boldsymbol{Q} = \{\boldsymbol{q}_n\}_{n=1}^{N}$ and reference video feature $\boldsymbol{V} = \{\boldsymbol{v}_i\}_{i=1}^{T}$. N and T are the length of the query and reference video (see Section for details). The backbone consists of:

Query Encoder. We use the same encoder as prior work (Yu et al. 2018), which contains a stack of encoder blocks. The details of the encoder block are shown in Figure 4 (a). It has multiple conv-layers, layer-norm layers, self-attention layers, and feedforward layers. The output of the query encoder is $\tilde{\boldsymbol{Q}} = \{\tilde{\boldsymbol{q}}_n\}_{n=1}^{N}$, which encodes the context in query.

Reference Encoder. The design of reference encoder is identical to the query encoder, and the output of this encoder is $\tilde{\boldsymbol{V}} = \{\tilde{\boldsymbol{v}}_i\}_{i=1}^{T}$, which encodes the context in video.

Query-Ref Co-Attention. It contains a co-attention mechanism to fuse the query and reference video features. Specifically, it firstly calculates a similarity matrix $\boldsymbol{S} \in \mathbb{R}^{T\times N}$, where each element $\boldsymbol{S}_{ij}$ denotes the similarity between $\tilde{\boldsymbol{v}}_i$ and $\tilde{\boldsymbol{q}}_j$. Thus, we obtains two weighted features $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$\boldsymbol{A} = \bar{\boldsymbol{S}} \cdot \tilde{\boldsymbol{Q}}, \quad \boldsymbol{B} = \bar{\boldsymbol{S}} \cdot \bar{\bar{\boldsymbol{S}}}^{T} \cdot \tilde{\boldsymbol{V}}, \tag{1}$$

where $\bar{\boldsymbol{S}}$ and $\bar{\bar{\boldsymbol{S}}}$ are the row-wise and column-wise normalized matrix of S, respectively. Then it composes a new frame feature sequences with i-th position is $[\boldsymbol{v}_i, \boldsymbol{a}_i, \boldsymbol{v}_i \odot \boldsymbol{a}_i, \boldsymbol{v}_i \odot \boldsymbol{b}_i]$, where $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are i-th row of $\boldsymbol{A}$ and $\boldsymbol{B}$, $\odot$ is the element-wise multiplication, and $[,]$ is the vector concatenate operation. And it uses another stack of encoder blocks (Figure 4 (a)) to encoder these new frame features. The output is $\boldsymbol{H}_0 = \{\boldsymbol{h}_i^0\}_{i=1}^{T}$, $\boldsymbol{H}_0 \in \mathbb{R}^{T\times D}$, where $\boldsymbol{h}_i^0 \in \mathbb{R}^D$ is i-th frame feature which encoders the query gist. Different from the existing bottom-up models which directly feeds $\boldsymbol{H}_0$ into the head network, GDP has a Graph-FPN layer to refine the frame features $\boldsymbol{H}_0$. It is worth noting that our proposed GDP is agnostic to the backbone, *i.e.*, it can be seamlessly incorporated into any stronger backbone to boost performance.

Graph-FPN Layer

As shown in Figure 4 (b), the Graph-FPN layer contains four main steps to refine the query-ref frame features $\boldsymbol{H}_0$:

Build Pyramid Hierarchy. Taking $\boldsymbol{H}_0$ from backbone, we firstly build a pyramid $\{\boldsymbol{H}_1 \in \mathbb{R}^{T_1\times D}, \boldsymbol{H}_2 \in \mathbb{R}^{T_2\times D}, \boldsymbol{H}_3 \in \mathbb{R}^{T_3\times D}\}$ with gradually half decrease the temporal resolution, *i.e.*, $T_{i+1} = T_i/2$. The network which transforms $\boldsymbol{H}_i$ to $\boldsymbol{H}_{i+1}$, is also a stack of the encoder blocks (Figure 4 (a)) with an extra stride-2 conv-layer to decrease resolution.

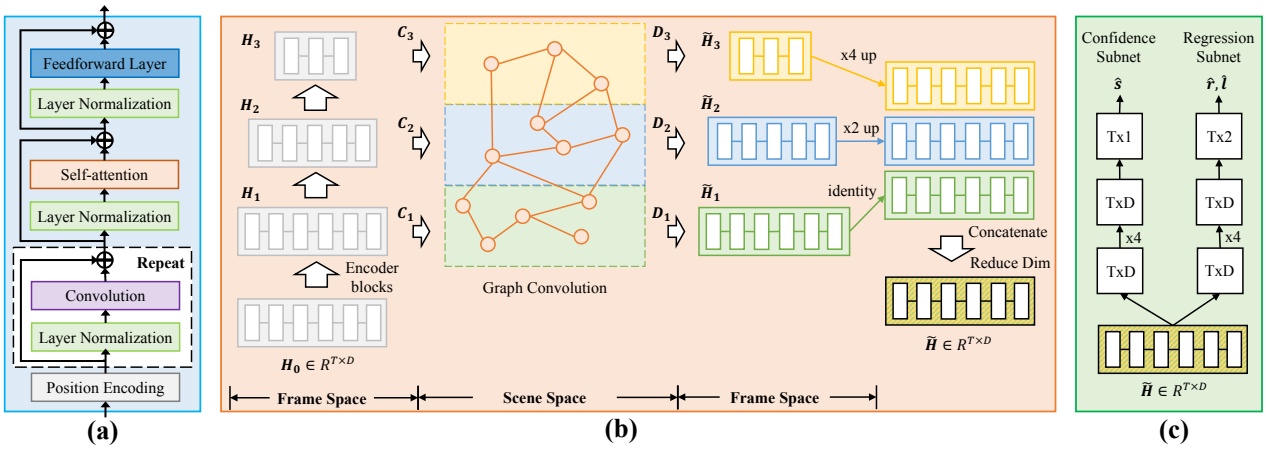

Figure 4: The details of each component of GDP. (a) The encoder block; (b) the Graph-FPN layer; (c) dense head network.

From Frame Space to Scene Space. After getting the multi-scale features $\{\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3\}$, we project them to a scene space. Taking $\boldsymbol{H}_2 = \{\boldsymbol{h}_i^2\}_{i=1}^{T_2}$ as an example, we aim to learn a scene-level features $\boldsymbol{X}_2 = f_2(\boldsymbol{H}_2) \in \mathbb{R}^{N_2 \times D}$, where N_2 is the number of node in scene space for this scale. We formulate the projection $f_2(\cdot)$ as a linear combination of origin features, *i.e.*, each node feature in scene space is:

$$\boldsymbol{x}_i^2 = \boldsymbol{c}_i^2 \boldsymbol{H}_2 = \sum\nolimits_j c_{ij}^2 \boldsymbol{h}_j^2, \tag{2}$$

where $\boldsymbol{C}_2 = [\boldsymbol{c}_1^2, ..., \boldsymbol{c}_{N_2}^2]$, $\boldsymbol{C}_2 \in \mathbb{R}^{N_2 \times T_2}$. $\boldsymbol{C}_2$ is derived from $\boldsymbol{H}_2$ through a 1×1 convolution. Similarly, we obtain $\boldsymbol{X}_1 \in \mathbb{R}^{N_1 \times D}$, $\boldsymbol{X}_3 \in \mathbb{R}^{N_3 \times D}$ from $\boldsymbol{H}_1$, $\boldsymbol{H}_3$ respectively.

Graph Convolution in Scene Space. After projecting the multi-scale features into the scene space, we adopt graph convolution (Kipf and Welling 2017) to exploit the scene relationships. In particular, we treats all N_{total} (*i.e.*, $N_{total} = N_1 + N_2 + N_3$) scene node features as a fully-connected graph, and the graph convolution is formulated as:

$$\boldsymbol{Y} = ((\boldsymbol{I} - \boldsymbol{A}_{adj})\boldsymbol{X})\boldsymbol{W}, \tag{3}$$

where $\boldsymbol{X} = [\boldsymbol{X}_1; \boldsymbol{X}_2; \boldsymbol{X}_3] \in \mathbb{R}^{N_{total} \times D}$ is the feature of all nodes in scene space, [;] is the row concatenate operation in matrix, $\boldsymbol{W} \in \mathbb{R}^{D \times D}$ is a transformation matrix, and $\boldsymbol{A}_{adj}$ is a learnable $N_{total} \times N_{total}$ node adjacency matrix. $\boldsymbol{I}$ denotes the identity connection to relief the optimization difficulties.

From Scene Space to Frame Space. Given the updated node feature in scene space $\boldsymbol{Y} = [\boldsymbol{Y}_1; \boldsymbol{Y}_2; \boldsymbol{Y}_3]$, we reserve project them to the frame space. Taking $\boldsymbol{Y}_2$ as an example:

$$\tilde{\boldsymbol{h}}_i^2 = \boldsymbol{d}_i^2 \boldsymbol{Y}_2 = \sum\nolimits_j d_{ij}^2 \boldsymbol{y}_j^2, \tag{4}$$

where $\boldsymbol{D}_2 = [\boldsymbol{d}_1^2, ..., \boldsymbol{d}_{T_2}^2]$, $\boldsymbol{D}_2 \in \mathbb{R}^{T_2 \times N_2}$.To reduce the computation cost, we set $\boldsymbol{C}_i = \boldsymbol{D}_i^T$. After getting new frame sequences: $\{\tilde{\boldsymbol{H}}_1, \tilde{\boldsymbol{H}}_2, \tilde{\boldsymbol{H}}_3\}$, we upsize $\tilde{\boldsymbol{H}}_1$ and $\tilde{\boldsymbol{H}}_2$ to the same resolution as $\tilde{\boldsymbol{H}}_3$, and concatenate all frames and decrease the dimension to obtain final features $\tilde{\boldsymbol{H}} \in \mathbb{R}^{T_1 \times D}$.

Dense Head Network

Different from the head network in existing bottom-up models, GDP regards each frame falling in the ground truth segment as positive samples. For each frame, there are two subnets, which aims to predict the boundary distances and confidence scores. The details about the two subnets are:

Boundary Regression Subset. It regresses the distances from each frame to the ground truth segment bi-directional boundaries. As shown in Figure 4 (c). Taking $\tilde{\boldsymbol{H}}$ from preceding Graph-FPN layer, this subset applies four 1×3 conv-layers with D channels, each followed by ReLU activation, and followed by 1×3 conv-layer with 1 channels. Then a sigmoid activation is attached to output two predictions (*i.e.*, left and right). For this subset, we only assign regression targets for positive frames. In particular, for the frame at i-th position, if the ground truth segment range is (t_s, t_e) (*i.e.*, $t_s \le i \le t_e$), the regression targets are $t_i^* = (l_i^*, r_i^*)$:

$$l_i^* = i - t_s, \quad r_i^* = t_e - i, \tag{5}$$

where l_i^* and r_i^* denotes the distances from i-th frame to the left and right boundaries, respectively.

Confidence Subset. Although each frame has a prediction about the ground truth segment, the confidence of each prediction should be different. The intuition comes from that a frame near the boundary should be easier to predict the distance to the boundary than a far one. Therefore, to take both left and right predictions into consideration, we use the "centerness" as ground truth confidence of positive samples. For negative samples, we set the ground truth to 0:

$$s_i^* = \begin{cases} \dfrac{\min(l_i^*, r_i^*)}{\max(l_i^*, r_i^*)}, & t_s \le i \le t_e \\ 0. & i < t_s \text{ or } i > t_e \end{cases} \tag{6}$$

Training and Inference

Loss. Give the predictions from all frames $\{(\hat{t}_i, \hat{s}_i)\}_{i=1}^{T}$ and the corresponding ground truth $\{(t_i^*, s_i^*)\}_{i=1}^{T}$, the total training losses of the GDP is:

$$L = \frac{1}{T} L_{conf}(\hat{s}_i, s_i^*) + \frac{1}{T_p} \mathbf{1}_{\{s_i^* > 0\}} L_{reg}(\hat{t}_i, t_i^*), \tag{7}$$

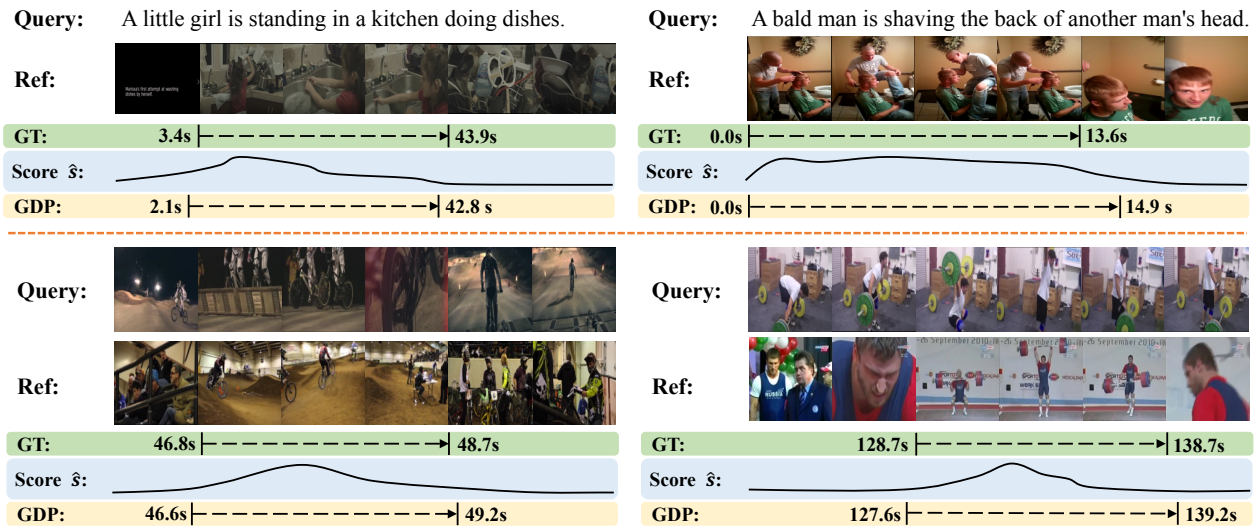

Figure 5: The qualitative results of GDP for NLVL on ActivityNet Captions (upper) and VRL on ActivityNet-VRL (below).

where T and T_p are the number of samples and positive samples. $\mathbb{1}_{\{s_i^*>0\}}$ is the indicator function, being 1 if $s_i^* > 0$ (*i.e.*, i-th frame is a positive frame) and 0 otherwise. L_{conf} is a binary cross entropy loss for the confidence subset. $L_{reg}(\hat{t}_i, t_i^*) = L_{l1}(\hat{t}_i, t_i^*) + L_{IoU}(\hat{t}_i, t_i^*)$ is the loss for boundary regression subnet, where L_{l1} is a smooth l_1 loss and L_{IoU} is an IoU loss (*i.e.*, $-ln\frac{\min(\hat{r}_i, r_i^*) - \max(\hat{l}_i, l_i^*)}{\max(\hat{r}_i, r_i^*) - \min(\hat{l}_i, l_i^*)}$).

Inference. At the test stage, we can obtain a confidence score and two boundary predictions from each frame. A straightforward solution is selecting the boundary predictions from the frame with highest confidence score. However, we empirically found the predictions from a single frame are usually with high variance. To relieve this situation, we follow Lu *et al.* (Lu et al. 2019) and use a **Temporal Pooling** to consider multiple frame predictions.

Experiments

Datasets

Natural Language Video Localization. We evaluated GDP on three prevailing NLVL benchmarks: 1) **TACoS** (Regneri et al. 2013): It consists of 127 videos and 17,344 text-to-clip pairs. The average duration of each video is 5 minutes. We used the standard split as (Gao et al. 2017), *i.e.*, 50% for training, 25% for validation, and 25% for test. 2) **Charades-STA** (Gao et al. 2017): It consists of 12,408 text-to-clip pairs for training, and 3,720 pairs for test. The average duration of each video 30 seconds. 3) **ActivityNet Captions** (Krishna et al. 2017): It is the largest NLVL benchmark with much more diverse context. Specifically, it consists of 19,209 videos and the average duration of each video is 2 minutes. We used the standard split as (Yuan, Mei, and Zhu 2019), *i.e.*, 37,421 text-to-clip pairs for training, and 17,505 pairs for test.

Video Relocalization. We evaluated GDP on the challenging VRL benchmark: **ActivityNet-VRL** (Feng et al. 2018), which is the only open released VRL dataset so far. It reorganizes the action recognition dataset ActivityNet (Caba Heilbron et al. 2015), by randomly selecting 160 action classes for training, 20 classes for validation, and 20 classes for test. This zero-shot split facilitates to evaluate the model generalization capability. For training, the query and reference video are randomly paired; for test, the pairs are fixed.

Evaluation Metrics

Natrual Language Video Localization. We evaluated models on two standard metrics: 1) **R@N, IoU@θ**: The percentage of test samples which have an IoU larger than threshold θ in one of the top-N predictions. Since the nature of bottom-up framework, we used N =1 in all experiments; 2) **mIoU**: The average IoU of top-1 predictions over all test samples.

Video Relocalization. We evaluated models on **mAP@1**, *i.e.*, the mAP of top-1 predictions over different thresholds.

Implementation Details.

Given a reference video $\mathcal{V}$, we first extracted the C3D features (Tran et al. 2015) of the down-sampled frames as the initial frame features, and utilized PCA to reduce the dimensions of these features to 500. For NLVL with language query $\mathcal{Q}$, we truncated or padded sentence to a maximum length of 15 words. Each word embedding was initialized with the 300-d Glove vector, and kept fixed all the time. Then we learned a transformation matrix to map these features into 500-d; For VRL with video query $\mathcal{Q}$, we followed the same processing steps as reference video. The dimension of all intermediate layers was set to 128. The node number N_1, N_2 and N_3 were set to 10. We trained the whole network from scratch with Adam optimizer for 100 epochs. The initial learning rate was set to 0.0001 and it was divided by 10 when the loss arrives on plateaus. The batch size of all experiments was set to 16, and the dropout rate was 0.5.

	Method	Venue	TACoS			Charades-STA			ActivityNet Captions		
			IoU@0.1	IoU@0.3	mIoU	IoU@0.3	IoU@0.5	IoU@0.7	IoU@0.3	IoU@0.5	mIoU
TD	VSA-RNN	*ICCV'17*	8.84	6.91	-	-	10.50	4.32	-	-	-
TD	VSA-STV	*ICCV'17*	15.01	10.77	-	-	16.91	5.81	-	-	-
TD	CTRL	*ICCV'17*	24.32	18.32	-	-	23.63	8.89	-	-	-
TD	ROLE	*MM'18*	-	-	-	25.26	12.12	-	-	-	-
TD	ACRN	*SIGIR'18*	24.22	19.52	-	-	-	-	-	-	-
TD	MCF	*IJCAI'18*	25.84	18.64	-	-	-	-	-	-	-
TD	TGN	*EMNLP'18*	-	-	-	-	-	-	43.81	27.93	-
TD	ACL	*WACV'19*	28.31	22.07	-	-	26.47	11.23	-	-	-
TD	SAP	*AAAI'19*	31.15	-	-	-	27.42	13.36	-	-	-
TD	QSPN	*AAAI'19*	-	-	-	**54.70**	35.60	15.80	45.30	27.70	-
RL	RWM	*AAAI'19*	-	-	-	-	36.70	-	-	36.90	-
RL	SM-RL	*CVPR'19*	26.51	20.25	-	-	24.36	11.17	-	-	-
BU	L-NET	*AAAI'19*	-	-	13.41	-	-	-	-	-	-
BU	ABLR-aw	*AAAI'19*	31.60	18.90	12.50	-	-	-	53.65	34.91	35.72
BU	ABLR-af	*AAAI'19*	34.70	19.50	13.40	-	-	-	55.67	36.79	36.99
BU	**GDP**	*AAAI'20*	**39.68**	**24.14**	**16.18**	54.54	**39.47**	**18.49**	**56.17**	**39.27**	**39.80**

Table 1: Performance compared with the state-of-the-art NLVL models on TACoS, Charades-STA and ActivityNet Captions.

Comparisons with State-of-the-Arts

Experiments on NLVL. We compared GDP with the state-of-the-art NLVL models. From the viewpoint of top-down and bottom-up frameworks, we group them into three categories: 1) Top-down models: **VSA-RNN**, **VSA-STV**, **CTRL** (Gao et al. 2017), **ROLE** (Liu et al. 2018c), **ACRN** (Liu et al. 2018b), **MCF** (Wu and Han 2018), **TGN** (Chen et al. 2018a), **ACL** (Ge et al. 2019), **SAP** (Chen and Jiang 2019), and **QSPN** (Xu et al. 2019); 2) RL-based models: **RWM** (He et al. 2019), **SM-RL** (Wang, Huang, and Wang 2019); 3) Bottom-up models: **L-Net** (Chen et al. 2019a), **ABLR-af**, **ABLR-aw** (Yuan, Mei, and Zhu 2019).

Results. The results on NLVL are reported in Table 1. We can observe that GDP achieves a new state-of-the-art performance over almost all evaluation metrics and benchmarks. It is worth noting that the performance gains in stricter metrics are more obvious (*e.g.*, 2.77% and 2.81% absolute improvement in mIoU on dataset TACoS and ActivityNet, 2.69% absolute improvement in IoU@0.7 on dataset Charades-STA).

Experiments on VRL. We compared GDP with the state-of-the-art VRL models. Similarly, we group them into two categories: 1) Top-down models: Frame-level and video-level baselines (Feng et al. 2018), **SST** (Buch et al. 2017); 2) Bottom-up models: **CGBM** (Feng et al. 2018).

Results. The results on VRL are reported in Table 2. We can observe that GDP surpasses all existing state-of-the-art models, especially for high-quality predictions (*e.g.*, GDP almost double the performance with tIoU threshold at 0.9).

mAP@1	0.5	0.6	0.7	0.8	0.9	Avg
Frame-level	18.8	13.9	9.6	5.0	2.3	9.9
Video-level	24.3	17.4	12.0	5.9	2.2	12.4
SST	33.2	24.7	17.2	7.8	2.7	17.1
CGBM	43.5	35.1	27.3	16.2	6.5	25.7
GDP	**44.0**	**35.4**	**27.7**	**20.0**	**12.1**	**27.8**

Table 2: Performance compared with state-of-the-art VRL models on ActivityNet-VRL.

Ablative Studies

Effectiveness of Graph-FPN Layer. To evaluate Graph-FPN layer, we designed three strong baselines. As shown in Figure 6, model A (a) is a bottom-up model with a backbone and a dense head network; model B (b) builds a pyramid hierarchy on top of backbone to capture higher semantics; model C (c) follows FPN (Lin et al. 2017) which combines adjacent scale features via a top-down connection; model D (d) is the GDP. In particular, we used the same backbone and the proposed dense head network in all four models.

Results. The results about the four models are reported in Table 3. We have the following observations: 1) Pyramid hierarchy is important for QBVL, *e.g.*, the model with pyramid (model B, C, and D) achieves better performance than the one without pyramid (model A); 2) Fusing only adjacent two scale features in FPN-style is not an effective solution to mitigate the semantic gaps between multi-scale features, *e.g.*, model C only gets comparable performance with model B. 3) Model D (*i.e.*, GDP) achieves the best performance over most of the metrics and benchmarks, which demonstrates the effectiveness of the Graph-FPN layer.

Effectiveness of Dense Predictions. To evaluate the dense predictions, we compared with a strong baseline which uses the same backbone and Graph-FPN layer as GDP. The only difference is replaceing the dense head network with the sparse head network (*i.e.*, directly predict boundaries).

Results. The results are reported in Table 4. We can observe that dense predictions significantly improve the performance over all benchmarks and metrics. In particular, the performance gap is more obvious in long video dataset (*e.g.*, TACoS). It demonstrates that dense predictions are beneficial to relieve the imbalance problem in the existing models.

Visualization

Qualitative Results. The qualitative results of GDP are illustrated in Figure 5. We can observe that the frame with the

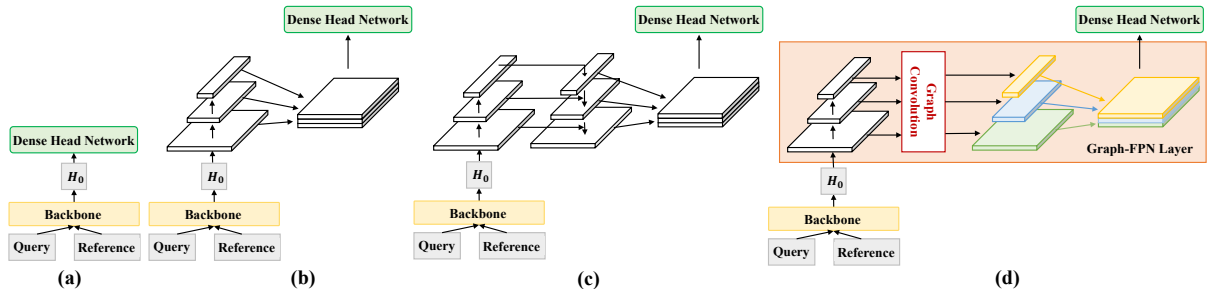

Figure 6: (a) Model A consists of a backbone and a dense head network. (b) Model B builds a pyramid hierarchy on top of backbone. (c) Model C uses FPN to combine two adjancent scale features. (d) Model D is the GDP with a Graph-FPN layer.

	NLVL												VRL				
	TACoS				Charades-STA				ActivityNet Captions				ActivityNet-VRL				
Model	IoU@			mIoU	IoU@			mIoU	IoU@			mIoU	mAP@1				
	0.1	0.3	0.5		0.3	0.5	0.7		01.	0.3	0.5		0.5	0.6	0.7	0.8	0.9
A	37.4	23.3	11.5	15.3	51.8	38.3	17.8	35.1	72.1	56.0	**40.7**	39.3	41.1	34.2	27.7	**20.3**	6.8
B	37.3	23.1	**13.9**	15.8	53.8	38.6	18.4	36.0	73.1	56.2	40.3	39.5	43.3	35.0	**27.9**	18.2	9.6
C	36.8	23.1	13.8	15.7	52.6	38.9	18.3	35.8	73.7	54.7	38.9	39.4	42.9	34.5	26.9	18.8	8.4
D	**39.7**	**24.1**	13.5	**16.2**	**54.5**	**39.5**	**18.5**	**36.6**	**75.0**	**56.2**	39.3	**39.8**	**44.0**	**35.4**	27.7	20.0	**12.1**

Table 3: Performance of different ablative models (model A, B, C, and D) on NLVL and VRL.

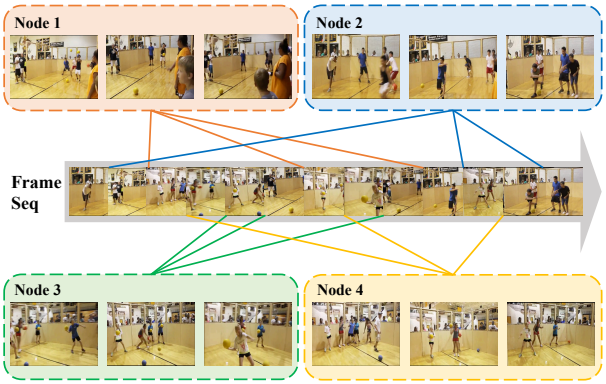

Figure 7: The visualization of nodes in the scene space.

highest score is always near the center of the ground truth segment, which conforms with our design, *i.e.*, using "centerness" as the confidence subnet targets.

Nodes in Scene Space. To visualize the nodes in scene space, we randomly select four nodes in the same scale and each node is represented by three video frames with highest attention weights. The results are illustrated in Figure 7. From the figure, we observe that each node in scene space is always a certain scene or with similar visual appearance.

Conclusion

In this paper, we thoroughly analyze the existing approaches for QBVL, especially the drawbacks of the current bottom-up framework. Based on the analysis, we proposed a novel bottom-up model GDP, which mitigates all the problems in existing bottom-up models: 1) It contains a Graph-FPN layer to encoder plentiful scene relationships and capture higher semantics; 2) It replaces the sparse boundary predictions with dense predictions to avoid the positive and negative samples imbalance. Extensive experiments on two QBVL tasks (NLVL and VRL) have demonstrated the effectiveness of GDP. Moving forward, we are going to design a hybrid model combining both top-down and bottom-up framework.

	Dataset	Metric	Head Network		
			Sparse	Dense*	Dense
NLVL	TACoS	IoU@0.1	32.3	36.5	**39.7**
		IoU@0.3	18.7	22.9	**24.1**
		IoU@0.5	9.6	13.0	**13.5**
		mIoU	12.9	15.2	**16.2**
	Charades-STA	IoU@0.3	52.9	53.9	**54.5**
		IoU@0.5	31.4	39.0	**39.5**
		IoU@0.7	14.7	18.3	**18.5**
		mIoU	35.1	36.1	**36.6**
	ActivityNet	IoU@0.1	72.4	73.5	**75.0**
		IoU@0.3	53.0	55.9	**56.2**
		IoU@0.5	37.5	**39.8**	39.3
		mIoU	39.0	39.3	**39.8**
VRL	ActivityNet	tIoU@0.5	41.6	42.3	**44.0**
		tIoU@0.6	30.5	35.3	**35.4**
		tIoU@0.7	25.7	27.6	**27.7**
		tIoU@0.8	19.8	**20.6**	20.0
		tIoU@0.9	8.5	**12.5**	12.1
		Average	25.2	27.7	**27.8**

Table 4: Performance compared with the model with sparse head network. * denotes model without temporal pooling.

Acknowledgement This work was supported by National Key Research and Development Program of China (2018AAA0101900), Zhejiang Natural Science Foundation (LR19F020002, LZ17F020001), National Natural Science Foundation of China (61976185, U19B200023, 61572431, U1611461), the Fundamental Research Funds for the Central Universities, Chinese Knowledge Center for Engineering Sciences and Technology, and Joint Research Program of ZJU & Tongdun Technology. Long Chen was supported by 2018 ZJU Academic Award for Outstanding Doctoral Candidates.

References

Anne Hendricks, L.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *ICCV*.

Buch, S.; Escorcia, V.; Shen, C.; Ghanem, B.; and Carlos Niebles, J. 2017. Sst: Single-stream temporal action proposals. In *CVPR*.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*.

Chen, S., and Jiang, Y.-G. 2019. Semantic proposal for activity localization in videos via sentence query. In *AAAI*.

Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; and Chua, T.-S. 2017. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*.

Chen, J.; Chen, X.; Ma, L.; Jie, Z.; and Chua, T.-S. 2018a. Temporally grounding natural sentence in video. In *EMNLP*.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018b. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*.

Chen, J.; Ma, L.; Chen, X.; Jie, Z.; and Luo, J. 2019a. Localizing natural language in videos. In *AAAI*.

Chen, L.; Zhang, H.; Xiao, J.; He, X.; Pu, S.; and Chang, S.-F. 2019b. Counterfactual critic multi-agent training for scene graph generation. In *ICCV*.

Chen, Y.; Rohrbach, M.; Yan, Z.; Shuicheng, Y.; Feng, J.; and Kalantidis, Y. 2019c. Graph-based global reasoning networks. In *CVPR*.

Feng, Y.; Ma, L.; Liu, W.; Zhang, T.; and Luo, J. 2018. Video re-localization. In *ECCV*.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. Tall: Temporal activity localization via language query. In *ICCV*.

Ge, R.; Gao, J.; Chen, K.; and Nevatia, R. 2019. Mac: Mining activity concepts for language-based temporal localization. In *WACV*.

He, D.; Zhao, X.; Huang, J.; Li, F.; Liu, X.; and Wen, S. 2019. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *AAAI*.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *ICCV*.

Law, H., and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *ECCV*.

Li, Y., and Gupta, A. 2018. Beyond grids: Learning graph representations for visual recognition. In *NeurIPS*.

Liang, X.; Hu, Z.; Zhang, H.; Lin, L.; and Xing, E. P. 2018. Symbolic graph reasoning meets convolutions. In *NeurIPS*.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *CVPR*.

Liu, A.-A.; Su, Y.-T.; Nie, W.-Z.; and Kankanhalli, M. 2016. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *TPAMI*.

Liu, A.; Xu, N.; Zhang, H.; Nie, W.; Su, Y.; and Zhang, Y. 2018a. Multi-level policy and reward reinforcement learning for image captioning. In *IJCAI*.

Liu, M.; Wang, X.; Nie, L.; He, X.; Chen, B.; and Chua, T.-S. 2018b. Attentive moment retrieval in videos. In *SIGIR*.

Liu, M.; Wang, X.; Nie, L.; Tian, Q.; Chen, B.; and Chua, T.-S. 2018c. Cross-modal moment localization in videos. In *ACM MM*.

Lu, C.; Chen, L.; Tan, C.; Li, X.; and Xiao, J. 2019. Debug: A dense bottom-up grounding approach for natural language video localization. In *EMNLP*.

Regneri, M.; Rohrbach, M.; Wetzel, D.; Thater, S.; Schiele, B.; and Pinkal, M. 2013. Grounding action descriptions in videos. *TACL*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *ICCV*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Wang, M.; Hong, R.; Li, G.; Zha, Z.-J.; Yan, S.; and Chua, T.-S. 2012a. Event driven web video summarization by tag localization and key-shot identification. *TMM*.

Wang, M.; Hong, R.; Yuan, X.-T.; Yan, S.; and Chua, T.-S. 2012b. Movie2comics: Towards a lively video content presentation. *TMM*.

Wang, M.; Luo, C.; Ni, B.; Yuan, J.; Wang, J.; and Yan, S. 2017. First-person daily activity recognition with manipulated object proposals and non-linear feature fusion. *TCSVT*.

Wang, W.; Huang, Y.; and Wang, L. 2019. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *CVPR*.

Wu, A., and Han, Y. 2018. Multi-modal circulant fusion for video-to-language and backward. In *IJCAI*.

Wu, C.-Y.; Feichtenhofer, C.; Fan, H.; He, K.; Krahenbuhl, P.; and Girshick, R. 2019. Long-term feature banks for detailed video understanding. In *CVPR*.

Xu, H.; He, K.; Plummer, B. A.; Sigal, L.; Sclaroff, S.; and Saenko, K. 2019. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*.

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

Yu, A. W.; Dohan, D.; Luong, M.-T.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Yuan, Y.; Mei, T.; and Zhu, W. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*.

Zhang, S.; Yan, S.; and He, X. 2019. Latentgnn: Learning efficient non-local relations for visual recognition. In *ICML*.

Zhou, X.; Wang, D.; and Krahenbuhl, P. 2019. Objects as points. In *arXiv*.

Zhou, X.; Zhuo, J.; and Krahenbuhl, P. 2019. Bottom-up object detection by grouping extreme and center points. In *CVPR*.