Polar Relative Positional Encoding for Video-Language Segmentation

Ke Ning[1] , **Lingxi Xie**[2] , **Fei Wu**[1] and **Qi Tian**[2]
[1]Zhejiang University
[2]Huawei Noah's Ark Lab
ningke@zju.edu.cn, 198808xc@gmail.com, wufei@zju.edu.cn, tian.qi1@huawei.com

Abstract

In this paper, we tackle a challenging task named video-language segmentation. Given a video and a sentence in natural language, the goal is to segment the object or actor described by the sentence in video frames. To accurately denote a target object, the given sentence usually refers to multiple attributes, such as nearby objects with spatial relations, *etc*. In this paper, we propose a novel *Polar Relative Positional Encoding* (PRPE) mechanism that represents spatial relations in a "linguistic" way, *i.e*., in terms of direction and range. Sentence feature can interact with positional embeddings in a more direct way to extract the implied relative positional relations. We also propose parameterized functions for these positional embeddings to adapt real-value directions and ranges. With PRPE, we design a *Polar Attention Module* (PAM) as the basic module for vision-language fusion. Our method outperforms previous best method by a large margin of 11.4% absolute improvement in terms of mAP on the challenging A2D Sentences dataset. Our method also achieves competitive performances on the J-HMDB Sentences dataset.

1 Introduction

In this paper, we tackle the video-language segmentation task. Given a video and a natural language description, the model is asked to generate pixel-level segmentation maps that segment the target object or the actor on interested frames according to the description. It is a very challenging task. On one hand, videos contain complex visual semantics. The semantics not only depends on a single frame but also different frames in the temporal domain. On the other hand, natural language sentences also imply complex logic relations. To accurately denote the target object, the description may need to use nearby objects and corresponding spatial relations. For example, as shown in Figure 1, the description "a girl in pink dotted dress is standing near the wall" describes the girl with the attributes "in pink dotted dress", "standing", and "near the wall". Among these attributes, "standing" is the action of this girl, "in pink dotted dress" and "near the wall" are both describing the spatial relations according to the dress and the wall. Therefore, the ability to exploit spatial relations is important to recognize the correct target.

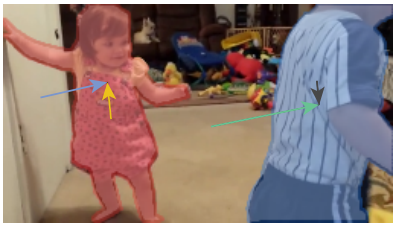

Figure 1: An illustration of the video-language segmentation task. Each description is referring different attributes and spatial relations. Spatial relations are highlighted in the sentences with corresponding arrows in the frame. Best viewed in color.

Recognizing objects and actions in videos has been widely researched. There are many existing methods for action recognition and localization [Carreira and Zisserman, 2017; Simonyan and Zisserman, 2014; Gu *et al.*, 2018] in videos. But these methods hardly model spatial relations between different objects. Non-local networks [Wang *et al.*, 2018] aggregate the relations between different parts on the image through self attention, and achieve great performances. But the plain self attention mechanism is positional agnostic. It only utilizes spatial relations in an implicit way.

Some other recent work [Shaw *et al.*, 2018; Huang *et al.*, 2019] tried to explicitly model relative positional relations on the feature maps. These methods define the relative positional embeddings on the feature grid. In the 2D image scenario, the differences of x and y coordinates are used to measure distances [Bello *et al.*, 2019]. But in most cases, natural language descriptions tend to describe the relations in terms of direction and range. For example, according to "a person on the left of the car", the target person has the direction relation "left", but without distance information. "Girl near the wall" implies the girl has a short distance to the wall. Therefore, in this paper, we propose a more direct relative positional encoding method that measures the spatial relations in terms of direction and range, a.k.a., in terms of polar, and define the corresponding embeddings to extract the implied spatial relations. Different from discrete coordinates, the direction ϕ and

range r are continuous real values. Therefore we also propose two functions to parameterize the direction and range embeddings respectively. As a side effect, the space complexity is much more effective. We denote this approach as Polar Relative Positional Encoding (PRPE). And with PRPE, we design our Polar Attention Module (PAM) as the basic vision-language fusion module in the network.

We evaluate our approach on two challenging datasets: A2D Sentences and J-HMDB Sentences. On A2D Sentences, our method outperforms the state-of-the-art method by a large margin of 11.4% absolute improvement in terms of mAP. Our method also achieves competitive performances on the J-HMDB Sentences dataset.

2 Related Work

2.1 Action Recognition and Localization in Videos

Action recognition is a fundamental research area in computer vision. Two-stream networks [Simonyan and Zisserman, 2014] and 3D ConvNets [Carreira and Zisserman, 2017; Tran *et al.*, 2015] are the most popular models for video feature learning. At finer granularities, temporal localization [Jiang *et al.*, 2014], spatio-temporal localization [Gu *et al.*, 2018] and segmentation [Perazzi *et al.*, 2016] are also important tasks for video analysis. There are also some recent work [Anne Hendricks *et al.*, 2017; Gao *et al.*, 2017] try to localize video clips temporally according to the given natural language description.

In this paper, we tackle the video-language segmentation task. The model needs to not only recognize the target object and its action but also extract visual information and relations described in the sentence.

2.2 Visual-Language Learning

Visual-language learning is a trending research direction in the area of machine learning. Many visual-language joint understanding tasks are gaining growing attention. Some of the most attractive tasks are visual question answering (VQA) [Antol *et al.*, 2015; Johnson *et al.*, 2017], referring expression localization [Hu *et al.*, 2016b], segmentation [Hu *et al.*, 2016a] and navigation [Anderson *et al.*, 2018], etc. A popular method to learn and exploit visual-language relation is the dynamic networks [Li *et al.*, 2017]. Compared to traditional visual models, dynamic networks generate visual filters as the network modules from the language input to transform visual features. In the context of video-language segmentation, [Gavrilyuk *et al.*, 2018] used language generated filter to transform visual feature maps in the upsampling stage.

2.3 Attention Mechanism and Positional Encoding

The attention mechanism, especially self attention, is an often-used mechanism in many machine learning areas, including computer vision [Wang *et al.*, 2018], natural language processing [Devlin *et al.*, 2019; Vaswani *et al.*, 2017], etc. The attention mechanism first computes an attention matrix, measures the correlation between each pair of input elements. According to the attention matrix, related elements are selected by weighted sum.

The naive attention mechanism does not encode position information. Two identical elements on different positions are considered representing same information, while in fact, they are not. Many positional encoding approaches are proposed to enhance the attention mechanism. The simplest way is to concatenate low-dimension coordinates onto the input features. Some other literature [Devlin *et al.*, 2019; Vaswani *et al.*, 2017] used positional embedding to represent position information. For each location, a distinct embedding is used to add with the input feature. The location information is embedded in the input feature for further modeling.

Another method to encode positional information is relative positional embedding. [Shaw *et al.*, 2018] introduced relative positional encoding in the scenario of 1D feature sequence. For each possible relative positional relation, it learns a distinct embedding vector. Besides from natural language processing, relative positional encoding had been also proved effective in many other areas [Huang *et al.*, 2019; Parmar *et al.*, 2019]. In the 2D scenario, [Bello *et al.*, 2019] applied 1D relative positional encoding on the x and y direction respectively.

3 Our Approach

3.1 Problem and Motivation

The video-language segmentation task receives an input video, $\mathcal{V} = \{f_i\}_{i=1}^{T}$, and a sentence, $\mathcal{S} = \{w_i\}_{i=1}^{L}$. For a subset of frames (of interest) in $\mathcal{V}$, $\{f_j\}$, the model is asked to generate a segmentation mask on each of these frames, which segments the object or actor that $\mathcal{S}$ describes. Please refer to Figure 1 for an example.

Video is a very complex data modality, especially in-the-wild videos. First, the scenes of video frames are various. There might be multiple similar objects within one frame. Second, videos consist of a series of continuous images. The images are evolving. Correctly understanding actions is still not an easy task. Therefore, to clearly denote a target object, the natural language descriptions usually refer to many attributes: the action, the position on the frame, and the relations with nearby objects.

In previous work for this task [Gavrilyuk *et al.*, 2018; Wang *et al.*, 2019], using the natural language description to generate dynamic network modules to transform visual features is a popular method. Dynamic networks have been proven effective in similar tasks. While the plain dynamic networks can only exploit the relations between the visual features and the linguistic features, *i.e.*, the object itself with its action. Position information is ignored in the process. Existing work about positional encoding [Shaw *et al.*, 2018; Bello *et al.*, 2019] are based on feature grids. For example, defining a distinct embedding for each position, or defining a distinct embedding for each row and column difference. With a proper positional encoding method, the absolute position and relative position can be more effectively utilized. Considering the relations are described by natural language sentences, which tend to describe spatial relations by words such as “left”, “top right”, “near” and “in”, etc, traditional grid-based positional encoding methods are not directly available to interact with language information.

In this paper, we propose a novel mechanism for explicitly encoding relative positional relations in terms of *direction* and *range*. We denote our method as Polar Relative Positional Encoding (PRPE). First, modeling relative positional relations enables our model to extract the spatial relations between objects on the frames. Second, modeling *direction* and *range* enables our model to exploit spatial relations described in natural language easier. In another word, our PRPE makes our model more "linguistic". With PRPE, we build our model and make use of spatial relations to tackle the video-language segmentation task.

3.2 Polar Relative Positional Encoding

Self Attention Review

Before introducing our PRPE, we would like to review the self attention mechanism first. Given a feature map B, we first compute a query Q and a key K using linear transformation, $Q = \mathrm{g}_q(B)$ and $K = \mathrm{g}_k(B)$, where g_q and g_k are distinct linear transformation functions. An attention matrix measures the similarities each pair of elements between Q and K, $A_{qk} = Q \cdot K$. Assuming B is a 2D feature map with shape $n \times n \times d$ (feature map size $n \times n$ and feature dimension d), the attention matrix A has a shape of $n^2 \times n^2$. $A_{i,j}$ represents the similarity or relation between the i-th feature in Q and the j-th feature in K. For each Q_i, A_i is aggregated using softmax, followed by weighted summing and linear transformation to compute the output value V:

$$V = \mathrm{g}_v(\mathrm{softmax}(A)B).$$

Polar Relative Positional Encoding

In the plain self attention mechanism, the relations between each pair of features are computed by the content similarity only. While in our task, we need to find out the object that the sentence describes. These sentences usually imply relative position relations. In order to capture the implicitly presented position information in the descriptions, we propose a novel mechanism of relative positional encoding that represents relative positional relations in the polar coordinate system. The polar coordinate system describes relative positions by direction ϕ and range r. By representing direction and range relations as vector-valued embeddings, the information implied by the given sentence $\mathcal{S}$ can be extracted and utilized directly.

Since ϕ and r are not discrete integers, these embeddings cannot be defined as discrete vector sets as before [Shaw *et al.*, 2018]. Therefore, we design two vector-valued functions f_ϕ and f_r to parameterize the direction and the range embeddings. The direction embedding function f_ϕ is only sensitive to the direction parameter ϕ. Absolute position and distance are irrelevant to f_ϕ. Similarly, the range embedding function f_r is only sensitive to the distance r. Absolute position and direction are irrelevant to f_r.

The range of ϕ is $[0, 2\pi)$. A direct thought of shaping embedding function $f_\phi(\phi)$ is using the trigonometric series, a.k.a., the Fourier series:

$$\mathrm{f}_\phi(\phi) = \mathbf{a}_{\phi,0} + \frac{1}{p}\sum_{i=1}^{p}(\mathbf{a}_{\phi,i}\cos i\phi + \mathbf{b}_{\phi,i}\sin i\phi),$$

where $\mathbf{a}_{\phi,i}$ and $\mathbf{b}_{\phi,i}$ are trainable coefficient vectors with dimension d.

Similarly, we use the same trigonometric series expansion for f_r to maintain numerical stability:

$$\mathrm{f}_r(r) = \mathbf{a}_{r,0} + \frac{1}{p}\sum_{i=1}^{p}(\mathbf{a}_{r,i}\cos\frac{2\pi ir}{r_{\max}} + \mathbf{b}_{r,i}\sin\frac{2\pi ir}{r_{\max}}),$$

where $\mathbf{a}_{r,i}$ and $\mathbf{b}_{r,i}$ are trainable coefficient vectors for f_r with dimension d, and $r_{\max}$ is the maximum possible range between two feature on a feature map. For a 2D feature map with size $n \times n$, $r_{\max} = \sqrt{2}(n-1)$. By training $\mathbf{a}_i$ and $\mathbf{b}_i$, f_ϕ and f_r can fit arbitrary vector-form function. f_ϕ and f_r contain totally $(4p+2)d$ trainable parameters.

With direction embedding function f_ϕ and range embedding function f_r, we can construct two $n^2 \times n^2 \times d$ tensors (vector-valued matrix), and extract the hidden feature representing the direction and range from the linear transformed sentence feature $\mathbf{s}$:

$$\begin{cases} A_\phi = \mathrm{g}_\phi(\mathbf{s}) \cdot \mathrm{M}(\mathrm{f}_\phi), & (1) \\ A_r = \mathrm{g}_r(\mathbf{s}) \cdot \mathrm{M}(\mathrm{f}_r). & (2) \end{cases}$$

A_ϕ is the weight matrix measures the direction relations described by $\mathcal{S}$, and A_r measures the range relations described by $\mathcal{S}$. M is the "Matrixize" operation that expands its parameter into the $n^2 \times n^2$ matrix form. Since f_ϕ and f_r are vector-valued functions, $\mathrm{M}(\mathrm{f}_\phi)$ and $\mathrm{M}(\mathrm{f}_r)$ are both 3D tensors with shape $n^2 \times n^2 \times d$. Together with A_{qk}, the final attention matrix is the sum of these three matrices:

$$A' = A_{qk} + A_\phi + A_r. \quad (3)$$

The final output of this module is the weighted sum according to the new attention matrix A':

$$V = \mathrm{g}_v(\mathrm{softmax}(A')B).$$

The sentence feature $\mathbf{s}$ is then used as the dynamic filter to transform visual features.

3.3 Full Model Architecture and Optimization

Full Model Architecture

Linguistic Encoder. We use a bi-LSTM to encode the input sentence. After the normal-order LSTM and the reverse-order LSTM, we concatenate the last hidden states of them as the representation of the input sentence $\mathbf{s}$.

Polar Attention Module and Full Model. With the mechanism we proposed before, we build our Polar Attention Module (PAM) and the full model. Figure 3 shows the architecture of our PAM. Our PAM takes a $t \times n' \times n' \times d$ 3D feature map B or a $n \times n \times d$ 2D feature map B and the sentence feature vector $\mathbf{s}$ as inputs. We first compress the input feature map into a "thumbnail" 2D feature map with shape $n \times n \times d$. We then transform B into Q' and K' using linear transform and channel-wise multiplication with $\mathbf{s}$. The reason we use channel-wise multiplication between Q', K' and $\mathbf{s}$ is we want to compute the $n^2 \times n^2$ attention matrix that measures feature relations not only by content. Instead, we use the sentence conditioned visual features, to measure the relations between visual features according to the given description.

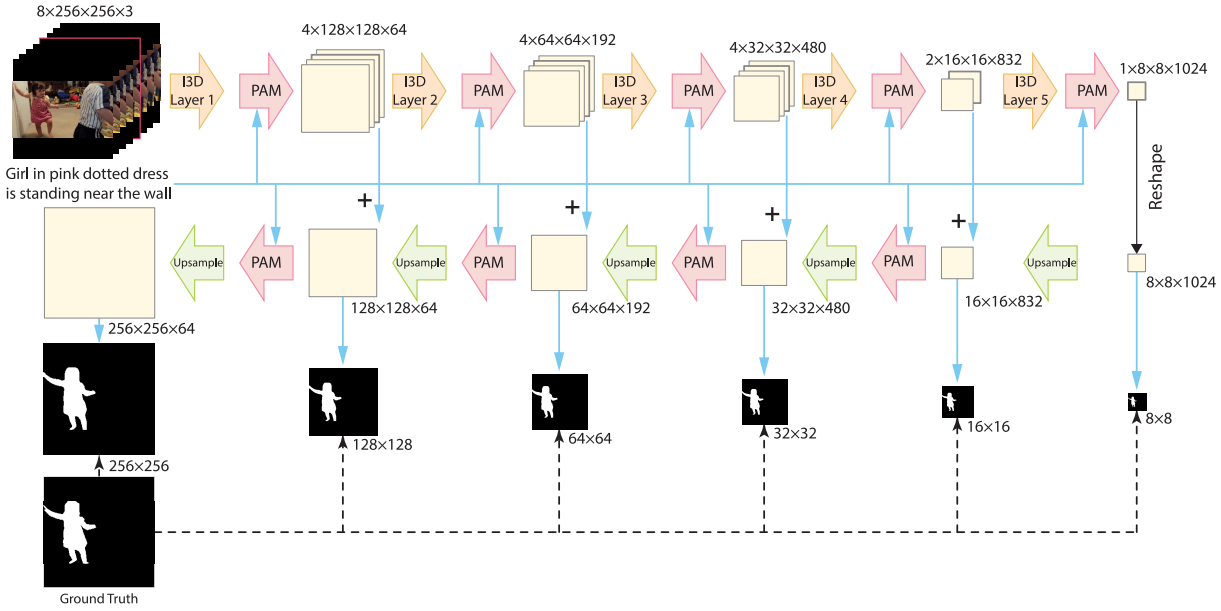

Figure 2: The architecture of our model. Red arrows, green arrows and orange arrows are our PAM, 2x expansion modules and pre-trained I3D modules, respectively. Blue arrows represent linear transformation. Best viewed in color.

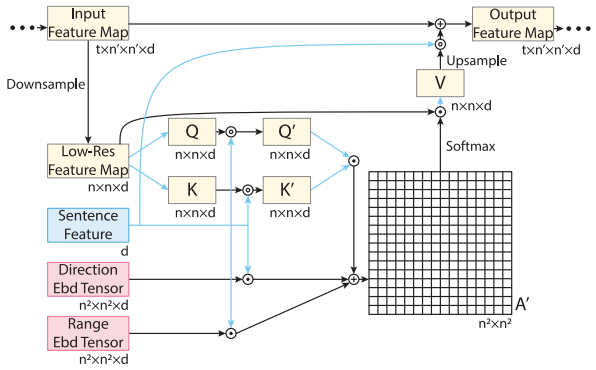

Figure 3: The architecture of our PAM. Blue arrows represent linear transformation. Filled and open co-centric circles represent inner product and channel-wise product respectively. Best viewed in color.

Both $\mathbf{a}_i$ and $\mathbf{b}_i$ are trainable parameters in the PAM. Together with $\mathbf{s}$, we obtain the final attention matrix A' following Equation 3 and the output value V. For the output value V, we also use $\mathbf{s}$ to transform it using channel-wise multiplication:

$$V' = \mathrm{g}_s(\mathbf{s}) \circ V,$$

where g_s is a linear transformation. We then interpolate V' to a feature map with shape same as B, and add it back into B.

Figure 2 shows our model's architecture. In the downsampling phase, we use a pre-trained I3D network as the backbone. Following the suggestion of [De Vries *et al.*, 2017], we insert our PAM between the Inception blocks, enabling language information works as a visual prior. During the upsampling phase, we apply our PAM and the 2x expansion alternatively. We add the feature map with the corresponding feature map from the downsampling phase to maintain frame details. At each step, we also generate an intermediate response map to be supervised by the ground truth. It is very helpful to keep the training stable. We use the average binary cross entropy over response map pixels as the loss function.

Optimization

In the actual implementation, we do not compute the entire $n^2 \times n^2 \times d$ tensor for direction and range embeddings. Computing the entire tensor is very space inefficient. Instead, we compute one $n^2 \times n^2$ matrix between $\mathbf{s}$ and each component $\mathbf{a}_i$ and $\mathbf{b}_i$ first:

$$\begin{cases} A_{\phi,a,i} = (\mathrm{g}_\phi(\mathbf{s}) \cdot \mathbf{a}_{\phi,i})\mathrm{M}(\cos i\phi), \\ A_{\phi,b,i} = (\mathrm{g}_\phi(\mathbf{s}) \cdot \mathbf{b}_{\phi,i})\mathrm{M}(\sin i\phi), \\ A_{r,a,i} = (\mathrm{g}_r(\mathbf{s}) \cdot \mathbf{a}_{r,i})\mathrm{M}(\cos \dfrac{2\pi i r}{r_{\max}}), \\ A_{r,b,i} = (\mathrm{g}_r(\mathbf{s}) \cdot \mathbf{b}_{r,i})\mathrm{M}(\sin \dfrac{2\pi i r}{r_{\max}}). \end{cases}$$

Hence, Equation 1 and Equation 2 are reformulated as:

$$\begin{cases} A_\phi = A_{\phi,a,0} + \dfrac{1}{p}\displaystyle\sum_{i=1}^{p}(A_{\phi,a,i} + A_{\phi,b,i}), & (4) \\ A_r = A_{r,a,0} + \dfrac{1}{p}\displaystyle\sum_{i=1}^{p}(A_{r,a,i} + A_{r,b,i}). & (5) \end{cases}$$

The M function expands $\sin$s and $\cos$s into a $n^2 \times n^2$ matrix. Since $\sin$s and $\cos$s here are scalar valued, M operations here only take $n^2 \times n^2$ spaces. The total space complexity is $n^2 \times n^2 \times (2p+1)$ for A_ϕ or A_r. A very small p can achieve pretty good performances, therefore the optimized version is much more efficient than the original $n^2 \times n^2 \times d$.

Methods	Precision@ 0.5	0.6	0.7	0.8	0.9	mAP 0.5:0.95	Overall IoU	Mean IoU
Baseline	54.2	47.8	36.6	21.2	3.6	29.9	61.7	45.5
Baseline + EF	59.7	52.5	40.5	22.9	3.6	32.7	63.0	48.7
Baseline 2	60.7	55.1	45.6	29.6	6.9	36.5	65.1	51.0
Baseline 2 + SA	60.3	54.4	45.7	30.0	7.3	36.6	65.0	50.7
Baseline 2 + SA + Gate	61.0	55.5	46.2	30.7	7.2	37.0	65.6	51.1
Baseline 2 + SA + Gate + xyEbd	62.9	57.2	48.2	31.8	7.3	38.3	**66.6**	**52.9**
Baseline 2 + SA + Gate + dxyEbd	62.0	56.6	46.5	31.4	7.9	37.8	66.1	52.0
Baseline 2 + SA + Gate + PRPE	**63.4**	**57.9**	**48.3**	**32.2**	**8.3**	**38.8**	66.1	**52.9**

Table 1: Ablation studies on the A2D Sentences datasets.

Methods	Precision@ 0.5	0.6	0.7	0.8	0.9	mAP 0.5:0.95	Overall IoU	Mean IoU
[Gavrilyuk *et al.*, 2018]	47.5	34.7	21.1	8.0	0.2	19.8	53.6	42.1
[McIntosh *et al.*, 2018]	52.6	45.0	34.5	20.7	3.6	30.3	56.8	46.0
[Wang *et al.*, 2019]	55.7	45.9	31.9	16.0	2.0	27.4	60.1	49.0
Ours	**63.4**	**57.9**	**48.3**	**32.2**	**8.3**	**38.8**	**66.1**	**52.9**

Table 2: Comparison with state-of-the-arts on the A2D Sentences dataset.

4 Experiments

4.1 Datasets and Implementation Details

A2D Sentences. A2D Sentences dataset is an extended version of the A2D dataset [Xu *et al.*, 2015]. There are 3,036 training videos and 746 testing videos. Each video has multiple different descriptions corresponding to different objects in the scenes. There are 5,359 training video-sentence pairs and 1,295 testing video-sentence pairs. The metrics to evaluate models for this dataset are precisions on different IoUs: 0.5, 0.6, 0.7, 0.8 and 0.9. In addition, mAP (mean average precision), overall IoU and mean IoU are used for evaluation. Overall IoU measures the total intersection area of all test data over the total union area. Mean IoU measures average over the IoU of each test sample.

J-HMDB Sentences. J-HMDB Sentences is an extension of the J-HMDB dataset [Jhuang *et al.*, 2013]. It Contains 928 videos and 928 corresponding sentences. A2D Sentences is used as the training set of J-HMDB. J-HMDB Sentences uses the same evaluation metrics as A2D Sentences.

Implementation Details. We use TensorFlow to implement our model. p is set to 3 in our experiments. The learning rate is 0.0005. We use a stack of 8 256×256 RGB frames as the video input for a balanced performance and speed.

4.2 Ablation Studies

Improving the Model Architecture

We perform ablation studies in this part. First, we perform an ablation study on model architecture. We implement our **Baseline** model by imitating [Gavrilyuk *et al.*, 2018], which is removing early-stage fusion and the residual connection between the encoding phase and decoding phase. And in the PAM, the visual feature is channel-wise multiplied by the sentence feature without self attention. The second model is adding the channel-wise multiplication for video-language feature between the Inception blocks in the encoding phase, denoted as **Baseline + EF** in Table 1. Since we use pre-trained I3D weights as the initialization of our encoder, inserting modules between them is possible to break the architecture of the pre-trained model. The third model then incorporates the residual connections, denoting as **Baseline 2** in Table 1. The first part of Table 1 shows the result of this ablation study.

The early fusion of visual and linguistic features brings a big improvement in terms of P@0.5 of 5.5%. On the other hand, the residual connections further improve the model. In terms of more strict metrics, especially P@0.9, residual connections improves the accuracy from 3.6% to 6.9%, which is a 91.7% relative improvement.

Ablation Study on the Module Structure

Starting from **Baseline 2**, we perform the ablation study on the module structure. We first add the self attention into our PAM, denoted as **Baseline 2 + SA**. We then use the sentence conditioned feature to compute the attention matrix. This model is denoted as **Baseline 2 + SA + gate**.

Together with self attention and the gate, we test different types of positional encoding methods. First we try to use the popular positional encoding method: positional embedding. The positional embedding method defines an embedding for each distinct position and adds them to the input feature. Since the input for our model is a series of RGB frames with a very low feature dimension 3, we apply positional embedding in each module. To be memory efficient, we define the positional embeddings for each x and y position, instead of every distinct position. This model is denoted as **Baseline 2 + SA + Gate + xyEbd**.

We also test the relative positional encoding method in terms of x and y differences. We apply relative positional encoding in each PAM. This model is denoted as **Baseline 2 + SA + Gate + dxyEbd** in Table 1.

Lastly, we use PRPE as our full model. we denote this model as **Baseline2 + SA + Gate + PRPE** in Table 1.

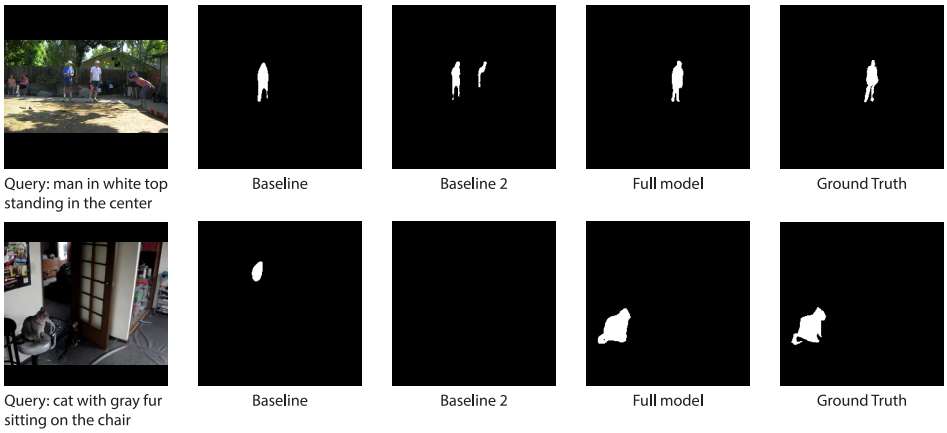

Figure 5: Qualitative analysis on the A2D Sentences dataset between our full model and the baseline models. Best viewed in color.

Figure 4: An example of our prediction result and the ground truth on the J-HMDB Sentences.

Methods	Precision@ 0.5	0.6	0.7	0.8	0.9	mAP 0.5:0.95
[Gavrilyuk *et al.*, 2018]*	69.9	46.0	17.3	1.4	0.0	23.3
[McIntosh *et al.*, 2018]	63.8	47.9	26.3	4.0	0.0	24.3
[Wang *et al.*, 2019]	**75.6**	56.4	28.7	3.4	0.0	28.9
Ours	69.07	**57.2**	**31.9**	**6.0**	**0.1**	**29.4**

Table 3: Comparison with state-of-the-arts on the J-HMDB Sentences dataset. * indicates the method used RGB+Flow visual input.

An interesting phenomenon is the plain self attention module does not improve the performances. A possible reason is the plain self attention cannot gather useful information by only measuring content similarities. By using sentence feature conditioned visual features to compute the attention matrix, the performances get small improvements.

Based on this model, we compare multiple positional encoding methods. All positional encoding methods improves a lot than the previous model. Our PRPE achieves the best performance, which improves 2.4%, 1.1% and 1.8% in terms of P@0.5, P@0.9 and mAP, respectively. It means our PRPE is effective to explore spatial relations. And spatial relations are very useful for high-quality segmentation.

4.3 Comparison with State-of-the-Arts

We first compare our approach with the SotA approaches on the A2D Sentences dataset. Compared to previous approaches, our approach achieves the best performances. On P@0.5, our method outperforms the SotA by a large margin of 7.7%. On P@0.9, our method outperforms the SotA by 4.7%, which is a 131% relative improvement.

We then compare our approach with the state-of-the-art approaches on the J-HMDB Sentences dataset. Our method outperforms previous methods on the metrics Precision@0.6 to Precision@0.9 and mAP. But our method achieves weak performances on Precision@0.5. An important reason is the ground truth mask from J-HMDB Sentences is not a standard segmentation map. The ground truth mask is generated from a puppet, which is not an accurate segmentation mask, as shown in Figure 4. Therefore our approach achieves lower results on certain metrics.

4.4 Qualitative Analysis

We show the qualitative analysis in Figure 5. In the first example, the description is looking for the "man in white top standing in the center". Both baseline models mistakenly segments the man on the left. With a correct understanding of attributes, our full model segments the correct person in white top. In the second example, the description is looking for the "cat with gray fur sitting on the chair". Both the **Baseline** model and the **Baseline 2** model failed to recognize the cat, while our model successfully located the gray cat on the chair. These examples showed that our approach takes advantage of spatial relations described by the sentence to improve segmentation.

5 Conclusions

We proposed a novel Polar Relative Positional Encoding mechanism along with a Polar Attention Module for video-language segmentation. Through extensive experiments, we proved the importance of spatial relations described in the sentence and the effectiveness of our proposed method. There are still many challenges remains in this task. Existing methods only considered a short snippet around the interested frames. Long-term action relations still remain unexplored. We leave this part as future work.

Acknowledgements

This paper is supported by NSFC (61625107, 61751209).

References

[Anderson *et al.*, 2018] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

[Anne Hendricks *et al.*, 2017] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017.

[Antol *et al.*, 2015] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

[Bello *et al.*, 2019] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *ICCV*, 2019.

[Carreira and Zisserman, 2017] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[De Vries *et al.*, 2017] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *NeurIPS*, 2017.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[Gao *et al.*, 2017] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017.

[Gavrilyuk *et al.*, 2018] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *CVPR*, 2018.

[Gu *et al.*, 2018] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.

[Hu *et al.*, 2016a] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016.

[Hu *et al.*, 2016b] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, 2016.

[Huang *et al.*, 2019] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *ICLR*, 2019.

[Jhuang *et al.*, 2013] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013.

[Jiang *et al.*, 2014] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014.

[Johnson *et al.*, 2017] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[Li *et al.*, 2017] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees GM Snoek, and Arnold WM Smeulders. Tracking by natural language specification. In *CVPR*, 2017.

[McIntosh *et al.*, 2018] Bruce McIntosh, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Multi-modal capsule routing for actor and action video segmentation conditioned on natural language queries. *arXiv preprint arXiv:1812.00303*, 2018.

[Parmar *et al.*, 2019] Niki Parmar, Prajit Ramachandran, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *NeurIPS*, 2019.

[Perazzi *et al.*, 2016] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[Shaw *et al.*, 2018] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL*, 2018.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.

[Tran *et al.*, 2015] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[Wang *et al.*, 2018] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[Wang *et al.*, 2019] Hao Wang, Cheng Deng, Junchi Yan, and Dacheng Tao. Asymmetric cross-guided attention network for actor and action video segmentation from natural language query. In *ICCV*, 2019.

[Xu *et al.*, 2015] Chenliang Xu, Shao-Hang Hsieh, Caiming Xiong, and Jason J Corso. Can humans fly? action understanding with multiple classes of actors. In *CVPR*, 2015.