Collaborative Spatial-Temporal Modeling for Language-Queried Video Actor Segmentation

Tianrui Hui[1,2*] Shaofei Huang[1,2,5*] Si Liu[3†] Zihan Ding[3] Guanbin Li[4,7]
Wenguan Wang[6] Jizhong Han[1,2] Fei Wang[5]
[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] Institute of Artificial Intelligence, Beihang University
[4] School of Computer Science and Engineering, Sun Yat-sen University
[5] SenseTime Research [6] Computer Vision Lab, ETH Zurich [7] Pazhou Lab, Guangzhou

Abstract

Language-queried video actor segmentation aims to predict the pixel-level mask of the actor which performs the actions described by a natural language query in the target frames. Existing methods adopt 3D CNNs over the video clip as a general encoder to extract a mixed spatio-temporal feature for the target frame. Though 3D convolutions are amenable to recognizing which actor is performing the queried actions, it also inevitably introduces misaligned spatial information from adjacent frames, which confuses features of the target frame and yields inaccurate segmentation. Therefore, we propose a collaborative spatial-temporal encoder-decoder framework which contains a 3D temporal encoder over the video clip to recognize the queried actions, and a 2D spatial encoder over the target frame to accurately segment the queried actors. In the decoder, a Language-Guided Feature Selection (LGFS) module is proposed to flexibly integrate spatial and temporal features from the two encoders. We also propose a Cross-Modal Adaptive Modulation (CMAM) module to dynamically recombine spatial- and temporal-relevant linguistic features for multimodal feature interaction in each stage of the two encoders. Our method achieves new state-of-the-art performance on two popular benchmarks with less computational overhead than previous approaches.

1. Introduction

Deep models have achieved notable progress in computer vision and other fields [10, 26, 24, 17]. Language-queried video actor segmentation [12] is an emerging task whose goal is to predict pixel-level mask for the actor performing some actions in a video described by a natural language query. Different from language-queried video spatial or temporal localization [44, 3, 1, 47], this task requires more fine-grained spatial-temporal modeling and visual-linguistic interaction to generate pixel-level prediction, thus is more challenging. At the intersection of computer vision and natural language processing [16, 18, 46, 25, 11, 31], this task enjoys a wide range of applications such as language-driven video editing [21], intelligent surveillance video processing [34] and human-robot interaction [29].

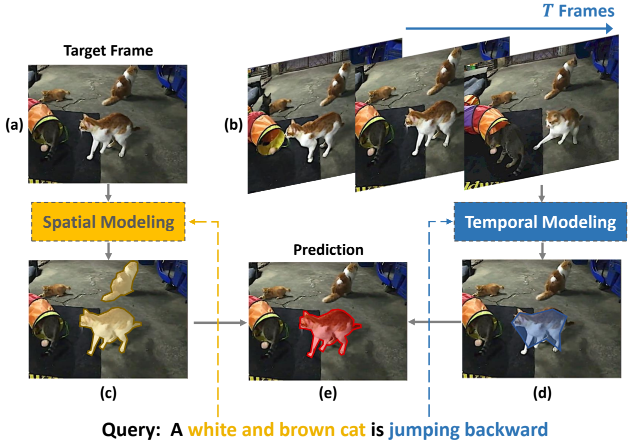

Figure 1. Illustration of our motivation. (a) The target frame. (b) The input video clip. (c) The spatial encoder can generate fine segmentation but may misidentify other actors due to weak action recognition ability. (d) The temporal encoder can recognize which actor is performing the queried action but may introduce misaligned spatial feature into the target frame, yielding inaccurate segmentation. (e) By integrating spatial and temporal encoders, the correct actor in the target frame can be well segmented.

[*] Equal contribution
[†] Corresponding author

As illustrated in Figure 1, given an input query "a white and brown cat is jumping backward" and an input video clip (we show 3 frames for brevity where the target frame is in the middle), language-queried video actor segmentation aims to segment the queried cat on the target frame. Since the output is based on the context of the whole video clip, we claim that both temporal modeling over the video clip and spatial modeling over the target frame are essential to solve this task. On one hand, as there are two white and brown cats in the target frame, spatial modeling cannot identify the correct cat by exploiting only appearance information. It instead inclines to producing fine but false-positive predictions on other cats. Therefore, the queried action needs to be recognized by incorporating information from adjacent frames to distinguish the jumping cat from the sitting one, leading to the necessity of temporal modeling over the video clip. On the other hand, the jumping cat has various poses and locations in 3 frames. Features of these spatially-misaligned pixels from adjacent frames will disturb the feature representation of the target frame during temporal modeling. The correspondence between the feature of the target frame and its ground-truth mask is hence broken. Thus, spatial modeling over the target frame is also necessary to provide precise spatial feature.

However, existing approaches [12, 39, 28, 38] conduct only temporal modeling over the video clip. Concretely, they first feed the video clip into a temporal encoder (3D CNN) to extract cross-frame video features, then apply temporal pooling over the time dimension to obtain a mixed feature of the target frame. As discussed above, mixing multi-frame spatial information will result in confused spatial feature of the target frame, leading to inaccurate segmentation. To tackle this limitation, we propose a collaborative spatial-temporal framework which contains two encoders to conduct spatial modeling over the target frame and temporal modeling over the video clip respectively. For the temporal encoder, we adopt a 3D CNN to identify the actor performing the queried action, which can be regarded as the coarse localization of the correct actor by temporal modeling. For the spatial encoder, we adopt a 2D CNN to extract precise spatial feature of the target frame, which serves as the fine segmentation of the correct actor by spatial modeling. To effectively integrate features from the two encoders, we introduce a Language-Guided Feature Selection (LGFS) module in the decoder to combine the two features with flexible channel selection weights, which are generated from the linguistic feature. Thus, language query serves as a selector to form comprehensive spatial-temporal feature for accurate segmentation.

In addition, language query contains both spatial-relevant information (appearance words, e.g., "white and brown") and temporal-relevant information (action words, e.g., "jumping backward"). When interacting with visual feature from the spatial encoder, features of spatial-relevant words should play a more important role than temporal-relevant words and vice versa. Therefore, we also propose a Cross-Modal Adaptive Modulation (CMAM) module which dynamically recombines linguistic features by cross-modal attention, yielding spatial- or temporal-relevant linguistic features to adaptively modulate corresponding visual features. By densely inserting our CMAM module into each stage of the two encoders, visual features can interact with linguistic features hierarchically and dynamically to highlight regions of the correct actor in spatial and temporal aspects.

The main contributions of our paper are summarized as follows: 1) We propose a collaborative spatial-temporal framework which contains a temporal encoder to recognize the queried action and a spatial encoder to generate accurate segmentation of the actor. A Language-Guided Feature Selection (LGFS) module is proposed in the decoder to aggregate spatial and temporal features comprehensively. 2) We also propose a Cross-Modal Adaptive Modulation (CMAM) module to conduct spatial- and temporal- relevant multimodal interaction dynamically in each stage of the two encoders. 3) Extensive experiments on two popular benchmarks show our method outperforms previous state-of-the-arts by large margins with 3× less computational overhead.

2. Related Work

2.1. Actor and Action Video Segmentation

To simultaneously infer various types of actors undergoing various actions, Xu *et al.* [43] collect a video dataset (A2D) in which both actors and actions in each video are annotated with pixel-level labels, and introduce a new task named actor and action video segmentation. They propose to formulate actors and actions using supervoxels and exploit a trilayer model to reason their relationships for joint labeling. Later in [42], they propose a graphical model to enable long-range interaction modeling among video parts. Yan *et al.* [45] explore the actor-action segmentation task in a weakly supervised setting with a multi-task ranking model. As deep models shown their powerful representation learning ability, Kalogeiton *et al.* [19] perform joint actor-action video segmentation via a "detection-segmentation" approach. Gavrilyuk *et al.* [12] further extend the A2D dataset with natural language descriptions and propose a new task called language-queried video actor segmentation. They exploit dynamic filters predicted from language features to convolve with video features and conduct segmentation based on the convolved heatmaps. Based on [12], Wang *et al.* [38] incorporate deformable convolutions [6] into dynamic filters to capture geometric variations. ACGA [39] explores co-attention mechanism between video and language features to extract multimodal

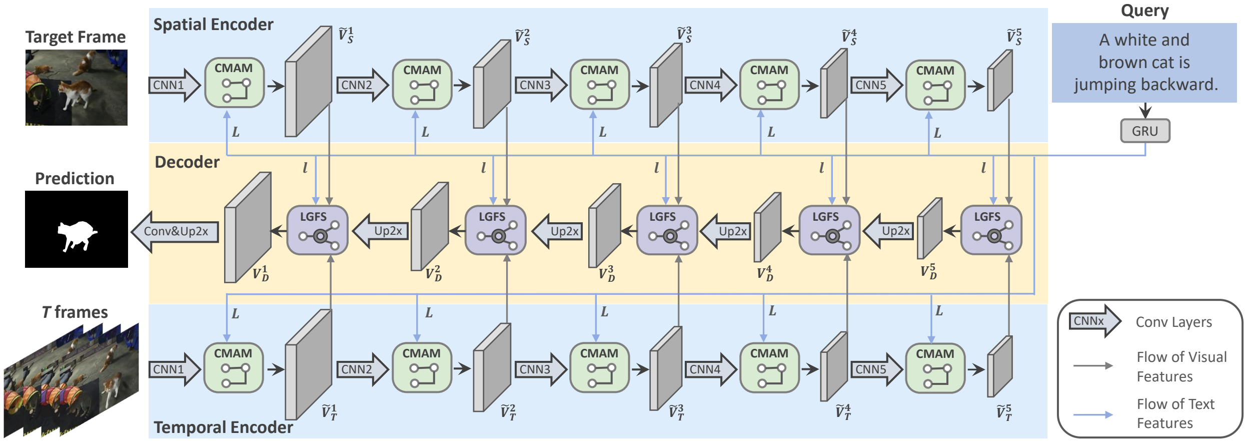

Figure 2. Overall architecture of our method. Spatial and temporal encoders extract features of the target frame and the video clip respectively, aided by CMAM which dynamically interacts multimodal features in each stage. LGFS is also densely applied in each stage of the decoder to flexibly fuse spatial and temporal features.

context for feature enhancement. Capsule networks [32] are exploited in [28] to encode video and language features for more effective representations that convolutions. Different from the above works which only use 3D CNNs to extract mixed spatio-temporal feature, we introduce a 2D spatial encoder to collaborate with original 3D temporal encoder for compensating the spatial misalignment brought by the temporal encoder.

2.2. Language-Queried Video Actor Localization

Some works have explored the alignment between visual and linguistic modalities by localizing actors and actions in the video by bounding boxes with language queries. For 1D temporal localization, Chen *et al.* [3] propose a cross-gated attended recurrent network to match video sequence and sentence, and perform cross-frame matching with a self-interactor. Local and global video feature integration is explored in [1] to match with input sentence more comprehensively. For 2D spatial localization, Yamaguchi *et al.* [44] first detect persons with spatio-temporal tubes and then conduct matching between tubes and textual descriptions. For 3D spatial-temporal localization, Chen *et al.* [4] propose a interesting task of localizing a spatial-temporal tube in the video corresponding to the given sentence in a weakly- supervised manner. Different from the above works, we identify the actors and actions more precisely with segmentation masks, providing fine-grained multimodal understanding.

2.3. Spatio-Temporal Modeling

Spatio-temporal modeling [33, 9, 40] is the key to solve video-related tasks. A direct way of spatio-temporal modeling is to use 3D CNNs such as C3D [36] and I3D [2], etc. To reduce the computational budget of 3D convolutions, (2+1)D ConvNet [37, 30] is proposed to decompose 3D convolution. SlowFast [8] proposes a slow path and a fast path to model spatial and motion information respectively. TSM [27] proposes a temporal shift module which shifts a portion of feature channels along the time dimension, and this operation can be regarded as a special case of 1D temporal convolution. In this paper, our model shares the same spirit with SlowFast where a spatial encoder and a temporal encoder are combined to collaboratively extract finer spatio-temporal context for pixel-level classification.

3. Method

The overall architecture of our method is illustrated in Figure 2. The input video clip and query are processed by visual and linguistic encoders respectively (i.e., CNNs [35] for video data and GRU [5] for sentence data). For visual modality, a spatial encoder and a temporal encoder are designed to extract spatial- and temporal-aware visual features respectively. In each stage of spatial and temporal visual encoders, visual features and linguistic features are fed into our proposed Cross-Modal Adaptive Modulation (CMAM) module to dynamically highlight visual features matched with the linguistic features. Then in the decoder, we propose a Language-Guided Feature Selection (LGFS) module to selectively fuse features of the spatial and temporal encoders from each stage. By progressive fusion and upsampling, our decoder produces a feature map of the same size as the input target frame to predict the segmentation mask.

3.1. Visual and Linguistic Encoders

Given a video clip with T frames where the target frame annotated with ground-truth mask is in the middle, we adopt Inception V3 [35] as the spatial encoder to process the target frame, and I3D [2] as the temporal encoder to process the whole video clip. We denote features of the i-th stage ($i \in [1,5]$) from the spatial and temporal visual encoders as $V_S^i \in \mathbb{R}^{H^i \times W^i \times C_V^i}$ and $V_T^i \in \mathbb{R}^{T^i \times H^i \times W^i \times C_V^i}$ respectively, where T^i, H^i, W^i, and C_V^i are the frame number, height, width and channel number of the i-th visual feature. We also adopt an 8-dimensional coordinate feature to encode relative position information of each pixel following [39]. Since the coordinate feature is densely fused with visual features in each stage of the encoders, we omit its denotation in the following formulas for ease of presentation. For the input textual query with N words, we utilize GRU [5] to extract the linguistic feature which is denoted as $L \in \mathbb{R}^{N \times C_L}$ where C_L denotes the channel number.

3.2. Cross-Modal Adaptive Modulation

Our CMAM aims to enable visual and linguistic features to interact with each other for highlighting visual features which are matched with the corresponding linguistic clues. We insert the proposed CMAM module into each stage of the spatial and temporal encoders. To clearly elaborate the multimodal interaction process in CMAM, we take the i-th stage of our spatial encoder as an example and omit the superscript i for simplicity. As illustrated in Figure 3, given the visual feature $V_S \in \mathbb{R}^{H \times W \times C_V}$ of the target frame and the linguistic feature $L \in \mathbb{R}^{N \times C_L}$ of the sentence, we first conduct cross-modal attention between V_S and L to compute an attention map $A \in \mathbb{R}^{N \times HW}$ which measures the feature relevance between each word and the target frame. Concretely, V_S and L are first transformed to the same subspace by convolutions:

$$V_S' = Conv_{2d}(V_S), \tag{1}$$

$$L_S' = Conv_{1d}(L), \tag{2}$$

where $V_S' \in \mathbb{R}^{H \times W \times C_M}$, $L_S' \in \mathbb{R}^{N \times C_M}$. Then, V_S' is reshaped to $\mathbb{R}^{HW \times C_M}$ to match the matrix dimensions. We further perform matrix product between V_S' and L_S' to obtain attention map A as follows:

$$A = L_S' \otimes {V_S'}^T \tag{3}$$

where $\otimes$ denotes matrix product.

Here $A \in \mathbb{R}^{N \times HW}$ measures the relevance between each word and each spatial location. Then we add all the values on the HW dimension and normalize it as follows:

$$\begin{aligned} \omega &= \sum_{j=1}^{HW} A^j, \\ \tilde{\omega} &= Softmax(\frac{\omega}{\|\omega\|_2}), \end{aligned} \tag{4}$$

where $\|\cdot\|_2$ denotes the L_2 norm of a vector, $A^j \in \mathbb{R}^N$ is the feature relevance between the j-th spatial location and N words, and $\tilde{\omega} \in \mathbb{R}^N$ is the normalized global feature relevance between each word and the whole target frame. Therefore, we can use these N weights to linearly re-combine features of N words to attain adaptive sentence feature $l_S = \sum_{k=1}^{N}(\tilde{\omega}^k L^k) \in \mathbb{R}^{C_L}$ which contains more spatial information matched with the spatial feature V_S of the target frame.

Afterwards, a linear layer and $sigmoid$ function are adopted to transform l_S to $\mathbb{R}^{C_V}$ dimensions and generate channel-wise modulation weights $\tilde{l}_S \in \mathbb{R}^{C_V}$:

$$\tilde{l}_S = \sigma(Linear(l_S)), \tag{5}$$

where σ denotes $sigmoid$ function. Inspired by SENet [14], we multiply $\tilde{l}_S$ with feature of the target frame V_S to highlight sentence-relevant visual feature channels and add the modulated feature with original V_S to ease optimization:

$$\tilde{V}_S = V_S + V_S \odot \tilde{l}_S, \tag{6}$$

where $\odot$ denotes elementwise product, $\tilde{V}_S$ is the output of the CMAM module and serves as the input feature of the next stage in the spatial visual encoder. For the temporal visual encoder, the same operations are applied on all the T frames to highlight sentence-relevant temporal visual features in each stage.

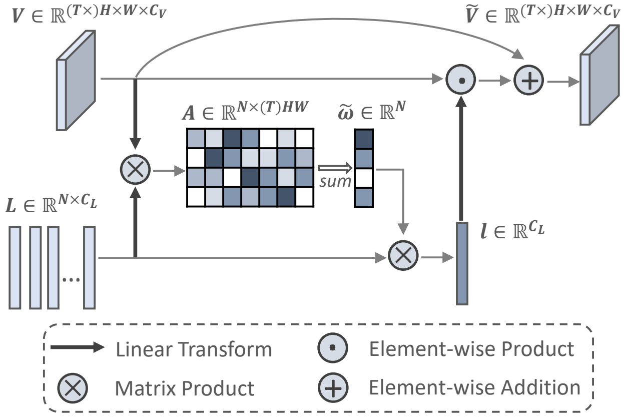
$V \in \mathbb{R}^{(T\times)H\times W\times C_V}$
$\tilde{V} \in \mathbb{R}^{(T\times)H\times W\times C_V}$
$A \in \mathbb{R}^{N\times(T)HW}$
$\tilde{\omega} \in \mathbb{R}^{N}$
sum
$L \in \mathbb{R}^{N\times C_L}$
...
$l \in \mathbb{R}^{C_L}$
Linear Transform
Element-wise Product
Matrix Product
Element-wise Addition

Figure 3. Illustration of CMAM module. Linguistic feature is dynamically recombined based on the relevance with visual feature. "S" and "T" subscripts in notations are omitted to denote general spatial or temporal features.

3.3. Language-Guided Feature Selection

In this section, we will elaborate our spatio-temporal decoder which progressively incorporates modulated features from spatial and temporal encoders to recover the feature resolution to the size of the input frame for mask prediction. The key problem is which encoder we should trust more for prediction during feature incorporation. To solve this problem, we propose a Language-Guided Feature Selection (LGFS) module to select the ratio of spatial and temporal features in the incorporation process under the guidance of linguistic feature. Our decoder contains 5 stages for consistency with the encoders and we take the i-th stage ($i \in [1, 5]$) as an example to detail our LGFS module.

Concretely, we conduct average pooling on the linguistic features $L \in \mathbb{R}^{N \times C_L}$ to obtain feature of the whole sentence $l \in \mathbb{R}^{C_L}$. Since l contains both spatial- and temporal-relevant linguistic information, we can use it to capture channel dependencies between spatial and temporal visual features inspired by SKNet [22], which proves to be effective on image classification. Two linear layers are applied on l to generate raw selection weights of each pair of channels from spatial and temporal visual features as follows:

$$g_S = Linear(l), \quad g_T = Linear(l), \tag{7}$$

where $g_S \in \mathbb{R}^{C_V}$ and $g_T \in \mathbb{R}^{C_V}$ have the same channel number with visual features. We apply $Softmax$ over each pair of channels of g_S and g_T to produce the normalized channel selection weights $\tilde{g}_S$ and $\tilde{g}_T$. The incorporated feature V_F^i is obtained as follows:

$$V_F^i = \tilde{V}_S^i \odot \tilde{g}_S + \tilde{V}_T^i \odot \tilde{g}_T, \tag{8}$$

where $\odot$ denotes elementwise product using broadcasting rule. We slightly abuse the notation of $\tilde{V}_T^i$ to denote the modulated feature of the target frame from the temporal encoder for simplicity. Finally, the output feature of the i-th stage in our decoder $V_D^i \in \mathbb{R}^{H^i \times W^i \times C_V^i}$ is defined as:

$$V_D^i = \begin{cases} V_F^i, & i = 5, \\ V_F^i + Upsample(V_D^{i+1}), & 1 \leq i \leq 4. \end{cases} \tag{9}$$

4. Experiments

4.1. Datasets and Evaluation Metrics

We conduct experiments on two popular language-queried video actor segmentation benchmarks including A2D Sentences [12] and J-HMDB Sentences [12]. Details of these two datasets are presented in the supplementary material due to limited space. We adopt Overall IoU, Mean IoU and Precision@X (P@X) as metrics to evaluate our model following prior works [39, 38]. Overall IoU calculates the ratio of the accumulated intersection area over the accumulated union area between predictions and ground-truth masks on all the test samples, while Mean IoU calculates the averaged IoU over all the test samples. Precision@X measures the percentage of test samples whose IoU are higher than a predefined threshold X, where $X \in [0.5, 0.6, 0.7, 0.8, 0.9]$. We also compute the Average Precision (AP) over the section of $[0.50 : 0.05 : 0.95]$.

4.2. Implementation Details

Following prior works [39, 38], we use the I3D [2] networks pretrained on Kinetics400 [2] dataset as our temporal visual encoder. For spatial visual encoder, we adopt Inception V3 [35] pretrained on ImageNet [7] dataset. We adopt two GRUs for the two visual encoders to extract linguistic features respectively. The maximum length of the input sentence is set as 20. We sample $T = 8$ RGB frames as the video input to our model where the annotated target frame is in the middle. The input frames are resized and padded to 320×320. Adam [20] is utilized as the optimizer and the training process is divided into two stages. First, we train the spatial and temporal networks (both encoders and decoders) respectively on A2D Sentence dataset for 12 epochs with batch size 8 and learning rate $5e^{-4}$ (divided by 10 every 8 epochs). Then, the two pretrained encoders are combined with a random-initialized decoder and fixed during finetuning the decoder for another 4 epochs with the same learning rate $5e^{-4}$. The details of loss functions for training are included in the supplementary material.

4.3. Comparison with State-of-the-arts

We conduct experiments on A2D Sentences and J-HMDB Sentences to compare our method with pervious state-of-the-arts. As illustrated in Table 1, our method remarkably outperforms pervious state-of-the-arts on A2D Sentences test set, indicating the effectiveness of collaborative learning of spatial and temporal encoders and adaptive visual-linguistic interaction. Comparing with CMDy [38], our method achieves 9.2% and 9.8% absolute improvements on precision metrics P@0.7 and P@0.8 respectively. **For the most rigorous metric P@0.9**, our method even **doubles** the previous performance of CMDy from 4.5% to 9.1%, demonstrating that our method can not only accurately identify the correct actor through cross-modal alignment, but also generate complete mask to cover the actor. Since Overall IoU favors large actors while Mean IoU treating actors of different scales equally, our improvements on IoU metrics also show that our method can well handle the scale variation of actors.

We further verify the generalization ability of our method on J-HMDB Sentences test set. Following prior works [38, 39], we use the best model pretrained on A2D Sentences dataset to directly evaluate all the test samples in J-HMDB Sentences without finetuning. For each testing video, 3 frames are uniformly sampled to evaluate the performance. As shown in the Table 2, our method accomplishes significant performance gains over previous state-of-the-arts, indicating that our method can excavate richer multimodal information through the mutual enhancement of spatial and temporal encoders, so as to obtain stronger generalization ability. Noted that all the methods including ours produce 0.0% on P@0.9, which is probably because mod-

Method	Precision					AP	IoU	
	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	0.5:0.95	Overall	Mean
Hu *et al.* [15] ECCV2016	34.8	23.6	13.3	3.3	0.1	13.2	47.4	35.0
Li *et al.* [23] CVPR2017	38.7	29.0	17.5	6.6	0.1	16.3	51.5	35.4
Gavrilyuk *et al.* [12] CVPR2018	47.5	34.7	21.1	8.0	0.2	19.8	53.6	42.1
Gavrilyuk *et al.* [12]† CVPR2018	50.0	37.6	23.1	9.4	0.4	21.5	55.1	42.6
ACGA [39] ICCV2019	55.7	45.9	31.9	16.0	2.0	27.4	60.1	49.0
VT-Capsule [28] CVPR2020	52.6	45.0	34.5	20.7	3.6	30.3	56.8	46.0
CMDy [38] AAAI2020	60.7	52.5	40.5	23.5	4.5	33.3	62.3	53.1
Ours	**65.4**	**58.9**	**49.7**	**33.3**	**9.1**	**39.9**	**66.2**	**56.1**

Table 1. Comparison with state-of-the-art methods on A2D Sentences test set. Our method significantly outperforms previous methods using only RGB input. † denotes utilizing additional optical flow input.

Method	Precision					AP	IoU	
	P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	0.5:0.95	Overall	Mean
Hu *et al.* [15] ECCV2016	63.3	35.0	8.5	0.2	0.0	17.8	54.6	52.8
Li *et al.* [23] CVPR2017	57.8	33.5	10.3	0.6	0.0	17.3	52.9	49.1
Gavrilyuk *et al.* [12] CVPR2018	69.9	46.0	17.3	1.4	0.0	23.3	54.1	54.2
Gavrilyuk *et al.* [12]‡ CVPR2018	71.2	51.8	26.4	3.0	0.0	26.7	55.5	57.0
ACGA [39] ICCV2019	75.6	56.4	28.7	3.4	0.0	28.9	57.6	58.4
VT-Capsule [28] CVPR2020	67.7	51.3	28.3	5.1	0.0	26.1	53.5	55.0
CMDy [38] AAAI2020	74.2	58.7	31.6	4.7	0.0	30.1	55.4	57.6
Ours	**78.3**	**63.9**	**37.8**	**7.6**	0.0	**33.5**	**59.8**	**60.4**

Table 2. Comparison with state-of-the-art methods on J-HMDB Sentences test set using the best model trained on A2D Sentences **without finetuning**. Our method shows notable generalization ability. ‡ denotes training more layers of I3D backbone on A2D Sentences.

els without training on J-HMDB Sentences cannot generate particularly fine masks on unseen samples.

4.4. Ablation Studies

We conduct ablation studies on the A2D Sentences dataset to evaluate different design of our framework.

Component Analysis. We summarize the ablation results of our proposed encoders and modules in Table 3a. The 1st and 2nd row denote our spatial- and temporal-only baselines where multimodal interactions only occurs in the decoders by visual and linguistic feature concatenation. Spatial and temporal baselines outperform each other on P@0.9 and P@0.5 respectively, which indicates spatial encoder can yield finer segmentation while temporal encoder can identify the actors more accurately. When simply combining the two encoders together in the 3rd row, we can observe an obvious performance boost in all metrics, which well demonstrates the complementarity of spatial and temporal encoders. Incorporating our LGFS module is able to further improve the performance, which shows that using language information as guidance can select effective spatial and temporal features more flexibly. In the 5th row, our CMAM module also brings large performance gain over a strong result in the 4th row. We insert CMAM into each stage of the spatial and temporal encoders to conduct multimodal feature interaction and remove the feature concatenation in our baselines. Results of CMAM prove the effectiveness of modulating visual features by dynamically recombine spatial- and temporal-relevant linguistic features.

Spatial-Temporal Feature Fusion. Table 3b presents results of different spatial-temporal feature fusion methods without CMAM module. Elementwise addition and maximization produce similar results. However, these simple fusion methods usually lack the ability to select appropriate spatial and temporal visual information according to the needs of language information. Our LGFS compensates for this deficiency and outperforms the above two operations, demonstrating the effectiveness of language guidance.

Inserting Positions of CMAM. We evaluate different inserting positions of CMAM and summarize the results in Table 3c. Inserting CMAM into the 5th and 4th stages of our spatial-temporal encoders can bring relatively significant improvements, which is probably because features from deep layers usually contain more high-level semantic information and are beneficial to the segmentation. As we insert CMAM into the shallow layers of encoders, segmentation performance is also constantly improved, showing the dynamically recombined linguistic features can modulate visual features of different abstraction levels.

Backbones. Table 3d shows the results of different

	Spatial	Temporal	LGFS	CMAM	Precision					AP	IoU	
					P@0.5	P@0.6	P@0.7	P@0.8	P@0.9	0.5:0.95	Overall	Mean
1	√				53.0	45.7	35.1	20.3	4.4	29.0	56.7	47.7
2		√			54.4	45.5	33.7	18.1	2.9	28.1	58.2	48.2
3	√	√			58.5	51.8	42.3	27.9	7.5	34.5	62.2	51.7
4	√	√	√		60.3	53.6	44.0	29.1	7.9	36.0	62.8	52.9
5	√	√	√	√	**65.4**	**58.9**	**49.7**	**33.3**	**9.1**	**39.9**	**66.2**	**56.1**

(a) **Component analysis.** Verifying the effectiveness of each component in our encoder-decoder framework. "Spatial" and "Temporal" denote spatial and temporal encoders respectively.

ST-Fusion	AP	IoU	
	0.5:0.95	Overall	Mean
Add	34.5	62.2	51.7
Max	34.8	62.6	51.6
LGFS (Ours)	**36.0**	**62.8**	**52.9**

(b) **Spaital and temporal feature fusion.**

Position	AP	IoU	
	0.5:0.95	Overall	Mean
$\{I_5\}$	36.9	63.1	53.3
$\{I_5, I_4\}$	38.5	65.2	55.0
$\{I_5, I_4, I_3\}$	39.0	65.4	55.6
$\{I_5, I_4, I_3, I_2\}$	39.4	65.9	55.7
$\{I_5, I_4, I_3, I_2, I_1\}$	**39.9**	**66.2**	**56.1**

(c) **Inserting positions of CMAM.** I_j denotes the j-th stage in Inception and I3D.

Backbone	AP	IoU	
	0.5:0.95	Overall	Mean
R50 + S3D	38.5	**67.1**	55.1
R50 [13] + I3D	39.5	66.4	**56.4**
IV3 + S3D [41]	38.3	66.7	55.0
IV3 + I3D	**39.9**	66.2	56.1

(d) **Backbones for our spatial and temporal encoder.** R50: ResNet-50. IV3: Inception V3.

Table 3. **Ablation studies**. Models are trained on A2D Sentences `train` split and evaluated on `test` split.

backbone selections for our spatial and temporal encoders. We conduct experiments on ResNet-50 [13] and S3D [41], which have similar capacities with their Inception and I3D counterparts. From the table we can observe that models with different backbones exhibit stable and high performances, which demonstrates that our collaborative spatial-temporal framework can well adapt to different backbones. For fair comparison with previous approaches, we adopt I3D and Inception V3 (which has moderate computational overhead comparing with ResNet-50) as our temporal and spatial encoders.

Method	Input Size	GFLOPs	AP
ACGA [39]	$16 \times 512 \times 512$	630.83	27.4
CMDy [38] †	$16 \times 512 \times 512$	> 600	33.3
Ours	$8 \times 320 \times 320$	213.06	**39.9**
Ours-Spa ‡	$8 \times 320 \times 320$	19.54	-

Table 4. Computational overhead analysis. The RGB input size is *Frames* × *Height* × *Width* (3 channels are omitted). † denotes the GFLOPs is estimated. ‡ denotes we only calculate the GFLOPs of our spatial encoder part, whose performance is not available.

4.5. Computational Overhead

We calculate the computational overhead of previous methods and ours in Table 4. Since CMDy has not released code, we estimate its computational overhead according to the details in their paper, including I3D backbone and the same input size as reported in the paper of ACGA. Both CMDy and ACGA rely on large input size to retain their performances. However, our method outperforms theirs with significant margins using 3× less GFLOPs and much smaller input size, showing that our method is more efficient in excavating useful information from multimodal features. Noted that our introduced spatial encoder only takes up 9.2% computational overhead of our full framework but brings considerable performance improvement, which well demonstrates the effectiveness of our collaborative spatial-temporal modeling framework.

4.6. Qualitative Analysis

Figure 4 presents the visualization results of our model on target frames, which provides qualitative analysis on the complementarity of our spatial and temporal encoders. As shown in the 2nd row, using only spatial encoder tends to make false-positive predictions on other actors (e.g., the man) which are irrelevant with the description due to the unawareness of action information, albeit the generated masks are relatively accurate. When only temporal encoder is used, the woman who is rolling can be correctly located but the segmentation result lacks some local details. For example, part of the woman's legs is misclassified as background. Incorporating both the spatial and temporal encoders yields precise segmentation on the correct actor. Similar phenomenon also appears in other examples of Figure 4.

We also visualize the attention maps between words and target frames in CMAM from spatial and temporal encoders in Figure 5. In the 1st row, spatial-relevant words "small" and "baby" yield highly-responsive attention maps on body of the two babies in the spatial encoder, while temporal-relevant word "crawling" mainly focuses on the moving hands and heads of the two babies in the temporal encoder. The word "running" in the 3rd row also focuses on the moving black dog to capture actions in the temporal encoder.

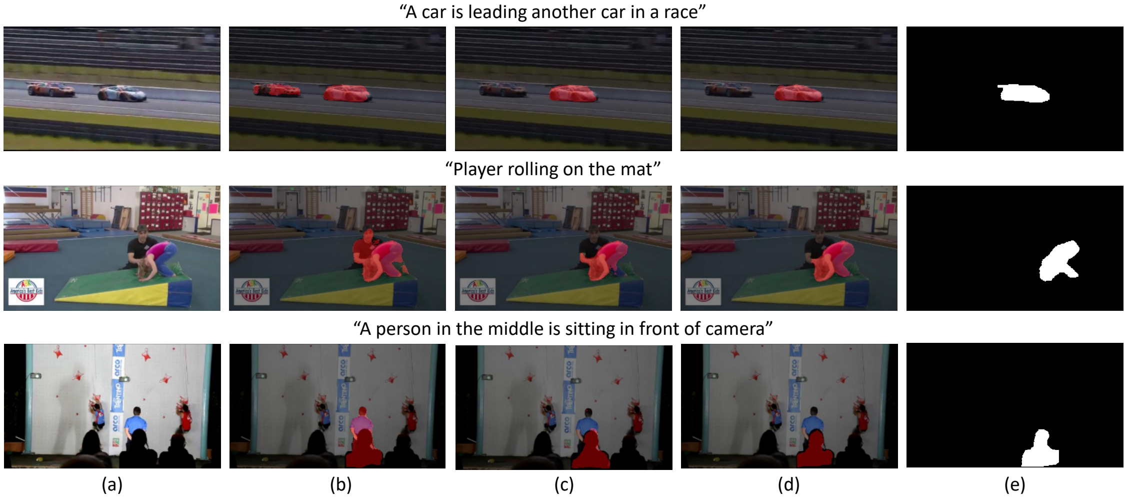
"A car is leading another car in a race"
"Player rolling on the mat"
"A person in the middle is sitting in front of camera"
(a)
(b)
(c)
(d)
(e)

Figure 4. Qualitative analysis on target frames. (a) Target frame. (b) Results of our model using spatial encoder only. (c) Results of our model using temporal encoder only. (d) Results of our model using spatial and temporal encoders. (e) Ground-truth.

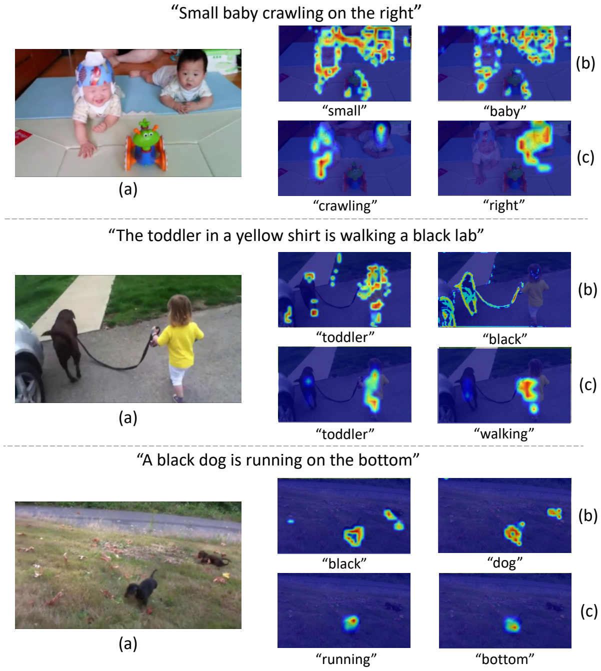
"Small baby crawling on the right"
"small"
"baby"
(b)
(a)
"crawling"
"right"
(c)
"The toddler in a yellow shirt is walking a black lab"
"toddler"
"black"
(b)
(a)
"toddler"
"walking"
(c)
"A black dog is running on the bottom"
"black"
"dog"
(b)
(a)
"running"
"bottom"
(c)

Figure 5. Visualization of attention maps between words and frames in CMAM. (a) Target frame. (b) Attention maps in spatial encoder. (c) Attention maps in temporal encoder.

These results show CMAM can well associate visual and linguistic features using spatial and temporal information.

Noted that for actor who only appears in a part of the video, our method can handle such case. For each frame, through our spatial and temporal encoders, intra-frame visual information and global cues from the whole video clip are collected and fused together. Then, the final segment per frame is guided by linguistic cues. Thus, our model does not explicitly leverage any causal information or frame-by-frame propagation. More experimental results are included in the supplementary material.

5. Conclusion and Future Work

In this paper, we explore the language-queried video actor segmentation task. We propose a collaborative encoder-decoder framework containing a 3D temporal encoder to recognize the queried actions and a 2D spatial encoder to well segment the actors, which alleviates the spatial misalignment issue brought by 3D CNNs in prior works. An LGFS module is introduced in the decoder to flexibly fuse spatial-temporal features. In addition, we also propose a CMAM module to dynamically recombine linguistic features for more adaptive multimodal feature interaction in each encoder. Our method outperforms previous methods by large margins on two popular benchmarks with 3× less computational overhead. In the future, we hope to accelerate current framework by designing more lightweight spatial and temporal encoders.

Acknowledgments This research is supported in part by National Natural Science Foundation of China (Grant 61876177, 61976250 and U1811463), Beijing Natural Science Foundation (4202034), Fundamental Research Funds for the Central Universities, the Guangdong Basic and Applied Basic Research Foundation (No. 2020B1515020048), Zhejiang Lab (No. 2019KD0AB04), SenseTime Group Ltd. and CCF-Baidu Open Fund.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 1, 3

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 3, 4, 5

[3] Jingyuan Chen, Lin Ma, Xinpeng Chen, Zequn Jie, and Jiebo Luo. Localizing natural language in videos. In *AAAI*, 2019. 1, 3

[4] Zhenfang Chen, Lin Ma, Wenhan Luo, and Kwan-Yee K Wong. Weakly-supervised spatio-temporally grounding natural sentence in video. *arXiv preprint arXiv:1906.02549*, 2019. 3

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 3, 4

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 3

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 3

[10] Chen Gao, Yunpeng Chen, Si Liu, Zhenxiong Tan, and Shuicheng Yan. Adversarialnas: Adversarial neural architecture search for gans. In *CVPR*, 2020. 1

[11] Chen Gao, Si Liu, Defa Zhu, Quan Liu, Jie Cao, Haoqian He, Ran He, and Shuicheng Yan. Interactgan: Learning to generate human-object interaction. In *ACM MM*, 2020. 1

[12] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *CVPR*, 2018. 1, 2, 5, 6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 4

[15] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016. 6

[16] Shaofei Huang, Tianrui Hui, Si Liu, Guanbin Li, Yunchao Wei, Jizhong Han, Luoqi Liu, and Bo Li. Referring image segmentation via cross-modal progressive comprehension. In *CVPR*, 2020. 1

[17] Shaofei Huang, Si Liu, Tianrui Hui, Jizhong Han, Bo Li, Jiashi Feng, and Shuicheng Yan. Ordnet: Capturing omnirange dependencies for scene parsing. *TIP*, 2020. 1

[18] Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In *ECCV*, 2020. 1

[19] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Joint learning of object and action detectors. In *ICCV*, 2017. 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H.S. Torr. Manigan: Text-guided image manipulation. In *CVPR*, 2020. 1

[22] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, 2019. 5

[23] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees GM Snoek, and Arnold WM Smeulders. Tracking by natural language specification. In *CVPR*, 2017. 6

[24] Yue Liao, Si Liu, Tianrui Hui, Chen Gao, Yao Sun, Hefei Ling, and Bo Li. Gps: Group people segmentation with detailed part inference. In *ICME*, 2019. 1

[25] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *CVPR*, 2020. 1

[26] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. In *CVPR*, 2020. 1

[27] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 3

[28] Bruce McIntosh, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Visual-textual capsule routing for text-based video segmentation. In *CVPR*, 2020. 2, 3, 6

[29] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, 2020. 1

[30] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 3

[31] Guanghui Ren, Lejian Ren, Yue Liao, Si Liu, Bo Li, Jizhong Han, and Shuicheng Yan. Scene graph generation with hierarchical context. *TNNLS*, 2020. 1

[32] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *NeurIPS*, 2017. 3

[33] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 3

[34] G. Sreenu and M. Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 2019. 1

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 3, 5

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

[37] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 3

[38] Hao Wang, Cheng Deng, Fan Ma, and Yi Yang. Context modulated dynamic networks for actor and action video segmentation with language queries. In *AAAI*, 2020. 2, 5, 6, 7

[39] 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. 2, 4, 5, 6, 7

[40] 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 *ECCV*, 2016. 3

[41] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 7

[42] Chenliang Xu and Jason J Corso. Actor-action semantic segmentation with grouping process models. In *CVPR*, 2016. 2

[43] Chenliang Xu, Shao-Hang Hsieh, Caiming Xiong, and Jason J Corso. Can humans fly? action understanding with multiple classes of actors. In *CVPR*, 2015. 2

[44] Masataka Yamaguchi, Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Spatio-temporal person retrieval via natural language queries. In *ICCV*, 2017. 1, 3

[45] Yan Yan, Chenliang Xu, Dawen Cai, and Jason J Corso. Weakly supervised actor-action segmentation via robust multi-task ranking. In *CVPR*, 2017. 2

[46] Tianyu Yu, Tianrui Hui, Zhihao Yu, Yue Liao, Sansi Yu, Faxi Zhang, and Si Liu. Cross-modal omni interaction modeling for phrase grounding. In *ACM MM*, 2020. 1

[47] 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 *CVPR*, 2019. 1