

Multi-Scale 2D Temporal Adjacent Networks for Moment Localization with Natural Language

Songyang Zhang, Houwen Peng, Jianlong Fu, Yijuan Lu, and Jiebo Luo

Abstract—We address the problem of retrieving a specific moment from an untrimmed video by natural language. It is a challenging problem because a target moment may take place in the context of other temporal moments in the untrimmed video. Existing methods cannot tackle this challenge well since they do not fully consider the temporal contexts between temporal moments. In this paper, we model the temporal context between video moments by a set of predefined two-dimensional maps under different temporal scales. For each map, one dimension indicates the starting time of a moment and the other indicates the duration. These 2D temporal maps can cover diverse video moments with different lengths, while representing their adjacent contexts at different temporal scales. Based on the 2D temporal maps, we propose a Multi-Scale Temporal Adjacent Network (MS-2D-TAN), a single-shot framework for moment localization. It is capable of encoding the adjacent temporal contexts at each scale, while learning discriminative features for matching video moments with referring expressions. We evaluate the proposed MS-2D-TAN on three challenging benchmarks, *i.e.*, Charades-STA, ActivityNet Captions, and TACoS, where our MS-2D-TAN outperforms the state of the art.



1 INTRODUCTION

TEMPORAL localization is a fundamental problem in video content understanding. There are several related tasks, such as temporal action localization [1], anomaly detection [2], video summarization [3], [4], and moment localization with natural language [5], [6]. Among them, moment localization with natural language is the most challenging one due to the complexity of moment description. This task is introduced recently by Gao *et al.* and Hendricks *et al.* [5], [6]. It aims to retrieve a temporary segment from an untrimmed video, as queried by a given natural language sentence. For example, given a query “a guy is playing the saxophone” and a paired video, the task is to localize the best matching moment described by the query, as shown in Fig. 1 (Query A). Video moment localization with natural language has a wide range of applications, such as video question answering [7], video content retrieval [8], dense video captioning [9], as well as video storytelling [10].

Most current language-queried moment localization methods follow a two-step pipeline [5], [6], [11]–[13]. It first utilizes sliding windows to generate moment segments from the input videos. Next, each moment is matched with the query sentence to estimate whether it is the target moment. This pipeline models different moments independently and neglects their temporal contexts. Thus it is difficult to precisely localize a moment that requires context information from other moments. For example, as shown in Fig. 1 (Query C), it targets to localize the query “the guy plays the saxophone again” in the video. If the model only watches the temporal moments in the latter part of the video, it cannot localize the described “again” moment accurately. Moreover, as shown in Fig. 1 (Query B), there are many temporal candidates overlapping with the target moment (the visualized time

slots). These candidates are relevant with respect to visual content, but they depict different semantics. Consequently, it may be difficult for existing methods to distinguish these visually similar moments since they process each temporal candidate individually.

To address these critical problems in moment localization with natural language, we propose a novel method, 2D Temporal Adjacent Networks (2D-TAN). The core idea is to localize a target moment from a two-dimensional temporal map, instead of a conventional one-dimensional sequential modeling as presented in Fig. 1. More specifically, for a temporal map with time unit τ , the (i, j) -th location on it represents a candidate moment starting from the timestamp $i\tau$ and lasting for $(j + 1)\tau$. This kind of 2D temporal map covers various video moments with different lengths, while representing their adjacent relations. In this fashion, 2D-TAN can perceive more moment context information when predicting whether a moment is related to other temporal segments. On the other hand, the adjacent moments in the map have content overlap but may depict different semantics. Considering them as a whole, 2D-TAN is able to learn discriminative features to distinguish them.

The time unit τ on the 2D temporal map decides the localization granularity. To guarantee a fine-grained localization of action instances, especially for precise temporal boundaries, we further build up multi-scale 2D temporal maps, in which multiple choices of time unit are considered. We construct K 2D temporal maps, each of which enumerates possible moments based on its time unit and longest duration. Such multi-scale modeling allows the localization networks to perceive moment candidates with more diverse temporal ranges and richer context. Moreover, the multi-scale modeling can be regarded as a sparse sampling with different intervals on the dense single-scale temporal map, thus could reduce the computation cost of moment generation. Compared with the dense single-scale case, sparse sampling reduces the total number of moment candidates from $O(N^2)$ to $O(N)$ (see section 4.5.1 for detailed proof),

- S. Zhang and J. Luo are with the Department of Computer Science, University of Rochester, Rochester, NY, 14627.
E-mail: szhang83, jluo@cs.rochester.edu
- H. Peng, J. Fu and Y. Lu are with Microsoft.
E-mail: {houwen.peng,jianf, yijlu}@microsoft.com

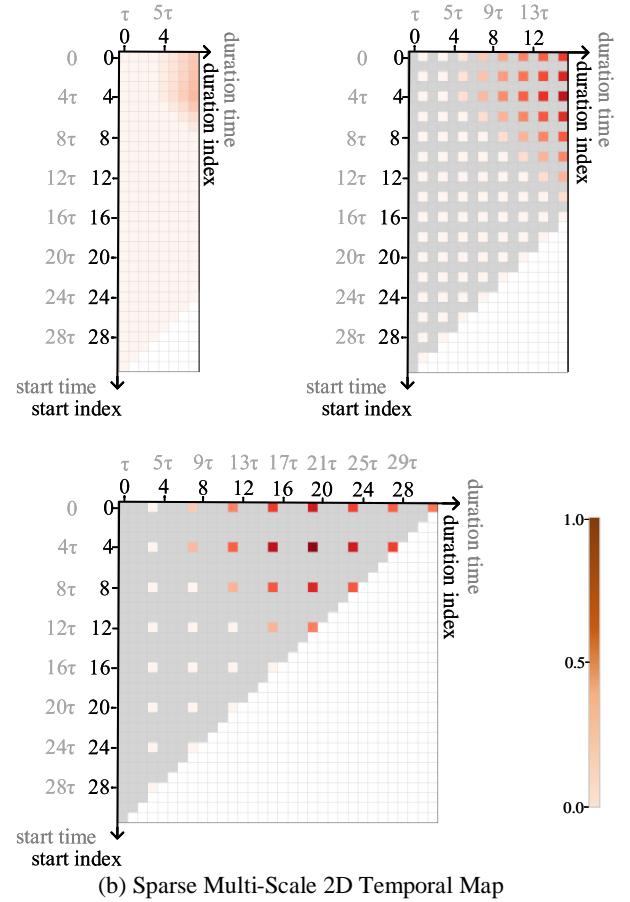
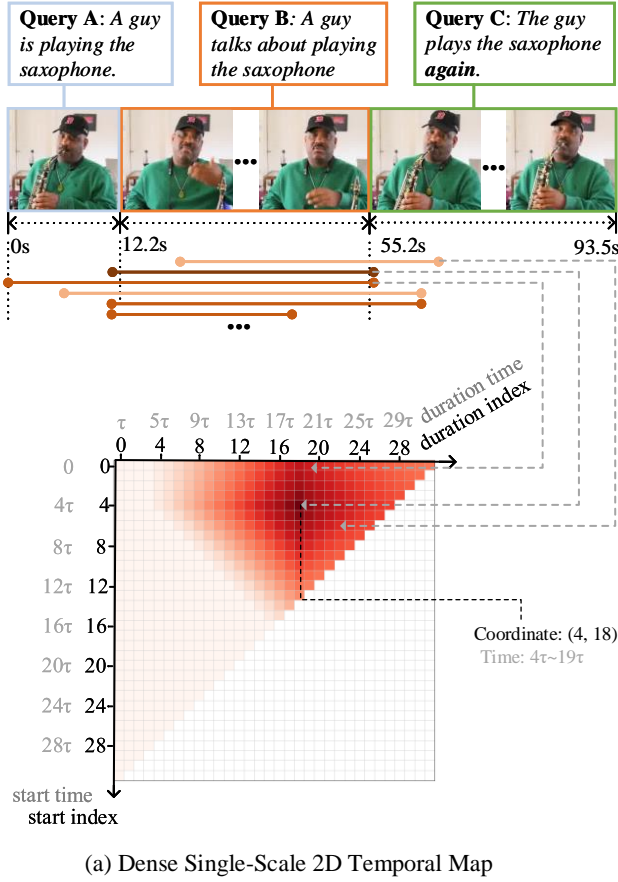


Fig. 1: Examples of the 2D temporal map. (a) Dense Single-scale 2D Temporal Map: the **black** vertical and horizontal axes represent the start and the end indices while the **gray** axes represent the corresponding start and end timestamp of a moment. The values in the 2D map, highlighted by **red** color, indicate the matching scores between the moment candidates and the target moment. Here, τ is a predefined short duration. The white boxes in the map indicate invalid moments. (b) Sparse Multi-Scale 2D Temporal Map: the sparse multi-scale 2D temporal map is composed of a series of 2D maps under different time units (τ , 2τ and 4τ in this figure). The color of boxes and axes follow the previous definition. In addition, the **gray** boxes on the 2D map represent the valid moments that are not selected. In this configuration, we are able to reduce the computational cost by modeling on smaller maps.

keeping a linear complexity in multi-scale modeling.

Some preliminary ideas of this paper have appeared in our earlier work [14]. In this paper, we extend the previously proposed single-scale 2D moment localization method to a multi-scale version. The new model considers the adjacent relations of moments at different temporal scales while achieving better performance with faster speed and less memory consumption. The main contributions of this paper are summarized as following:

- We introduce a novel *two-dimensional temporal map* for modeling the temporal adjacent relations of video moments. Compared with previous methods, the 2D temporal map enables the model to perceive more video context information and learn discriminative features to distinguish the moments with complex semantics.
- We propose a *Multi-Scale 2D Temporal Adjacent Network*, i.e., MS-2D-TAN, for moment localization with natural language. The multi-scale modeling allows our model to have a larger receptive field and obtain richer context. Meanwhile, it reduces the complexity of generat-

ing moments from quadratic to linear, which makes the dense video prediction more efficient. Without any sophisticated video-language cross-modality fusion, MS-2D-TAN achieves competitive performance in comparison with the state-of-the-art methods on three benchmark datasets. Our code and models are publicly available at <https://github.com/microsoft/2D-TAN>.

2 RELATED WORK

There are two major sub-fields in the temporal localization in untrimmed videos: temporal action localization and moment localization with natural language. Temporal action localization aims to predict the start and the end time and the category of an activity instance in untrimmed videos. The representative methods include two-stage temporal detection methods [1] and one-stage single shot methods [15]. This task is limited to pre-defined simple actions and cannot handle complex events in the real world. Therefore, moment localization with natural language [5], [6] is introduced recently to tackle this problem. In the remainder of this

section, we first review recent clip-based and moment-based methods, and then discuss their cross-modality alignment. In addition, we introduce some related methods in other tasks and their difference from our preliminary work.

Localizing moments in videos by query sentences is very challenging. It not only needs to understand video content, but also requires to align the semantics between video and language. A video clip is a set of short video segments with fixed duration and interval, while a moment is a sequence of video clips. Depending on whether the moment features are extracted, we divide current methods into two categories, clip-based methods (not extracted) and moment-based methods (extracted).

Clip-based methods. The main idea of clip-based methods is to align video clips with language directly and predict matching scores without extracting moment features. In general, there are three common ways to map a clip to a moment score: anchor-based methods, anchor-free methods, and RL-based methods. Anchor-based methods define a set of anchors with a fixed length for each clip [16]–[18], while anchor-free methods directly predict the start and the end time through classification [19], [20] or regression [12], [21]–[28]. RL-based methods model the task as a sequential decision-making problem and solve it by reinforcement learning [29]–[32]. In terms of their context modeling, most approaches [12], [16], [17], [20]–[22], [25] gradually aggregate the context information through a recurrent structure. Some approaches [18], [23], [29] model surrounding clips as the local context using 1D convolution layers, while other approaches model the entire clip as the global context through self-attention modules [19], [24], [26], [27]. Since clips are the shortest moments, the clip-level context is a subset of moment-level context. Thus, all these context modeling methods ignore a large part of moment-level context. In contrast, our focus is to model contexts on the moment level and to learn discriminative moment features to boost the localization performance.

Moment-based methods. Moment-based methods extract moment-level features and learn a matching function between features and the query. Most approaches predict moment scores in one stage, while some approaches follow a cascaded strategy and filter moments by multiple stages [33], [34]. Different from clip-based methods, existing moment-based methods integrate temporal context in two common ways. One way is to use the whole video as the global context. For example, Hendricks *et al.* [6] and He *et al.* [29] concatenate each moment feature with the global video feature [6] as the moment representation. Wang *et al.* concatenate semantic features with the global video feature [30]. Another way is to use the surrounding clips as the local context for a moment. Gao *et al.*, Liu *et al.*, Song *et al.* and Ge *et al.* concatenate the moment feature with clip features before and after current clip as its representation [5], [11]–[13]. Since these methods only consider one or two specific moments, the rich context information from other possible moments is ignored. One recent exploration [35] of aggregating the context from other moments is Graph Convolutional Network (GCN). However, one inherent feature of graph is *node permutation invariance* [36], switching any two nodes does not change the result. Therefore, the graph modeling ignores the temporal ordering of different

moments. In contrast, our method considers all the neighboring moments as the context and models the moments on a 2D convolution network. The 2D convolution network can naturally preserve the relative position of different moments. This design enables the model to perceive more context information and learn more discriminative features.

Video and language cross-modality alignment. The simplest way for cross-modal alignment is to directly multiply or concatenate the clip/moment features with sentence features [5], [6], [23]. Recent approaches improve video and language alignment from several different directions, such as cross-modal attention, sentence syntactic modeling, and compositional reasoning. The key idea of cross-modal attention is to attend relevant video clips/moments or query words from another modality. Some methods attend relevant video features through words [12], [28], while most other methods attend both the relevant video features and words via the co-attention module [13], [16], [17], [19], [20], [23], [25], [27], [28], [35], [37], [38]. For sentence syntactic modeling, Zhang *et al.* [17] enhance the sentence modeling with the queries’ syntactic graph. For compositional reasoning, Liu *et al.* [37] explicitly use the compositionality in natural languages for temporal reasoning in videos. Instead of using the complex attention modules, our proposed MS-2D-TAN model only adopts a simple multiplication operation for visual and language feature fusion.

Discussion with related methods in other tasks The self-attention operation in document modeling enumerate all possible word pairs. This enumeration is similar to our 2D temporal map, where the start and the end time pairs are enumerated. Due to the enumeration, both tasks face the same problem: memory cost and speed scale up quadratically with the sequence length. Several methods are recently proposed to improve the efficiency in the long document tasks. Beltagy *et al.* [39] combine global attention, dilated convolution and sliding window together to reduce the complexity to linear. Sukhbaatar *et al.* [40] design an adaptive attention span, which learns a hard mask to ignore the long-term relations. In this paper, we focus on reducing the computational cost in the video domain. Our study is also related to some contemporaneous and recent work [41], [42] that apply 2D temporal map to temporal action localization task. In contrast to their work, our work focus on moment localization with natural language. Besides, we also propose a multi-scale 2D temporal map to improve the efficiency and the performance of the dense 2D temporal map.

Difference from our preliminary work. Some preliminary ideas in this paper appeared in the conference version [14]. Compared with [14], the proposed MS-2D-TAN in this paper is more efficient, and subsumes the version in [14]. Our MS-2D-TAN not only inherits the major advantages of the preliminary model, *i.e.*, learning discriminative moment features through a 2D temporal map, but also further improves the efficiency by reducing the complexity from quadratic to linear. The experimental results (Sec. 4) show clearly that the new model is more efficient (Fig. 4) and achieves better performance in three benchmark datasets (Table. 2 3 4).

3 OUR APPROACH

In this section, we first introduce the problem formulation of moment localization with natural language. Next, we present the pipeline of our proposed network, including language representation, video representation and multi-scale 2D temporal adjacent network, as shown in Fig. 2. Finally, we present the training and the inference of our network.

3.1 Problem Formulation

Given an untrimmed video V and a sentence S as a query, the task is to retrieve the best matching temporary moment M specified by the query. More specifically, we denote the query sentence as $S = \{s_i\}_{i=0}^{l^S-1}$, where s_i represents a single word, and l^S is the total number of the words. The input video is a frame sequence, *i.e.* $V = \{x_i\}_{i=0}^{l^V-1}$, where x_i represents a frame in a video and l^V is the total number of frames. The retrieved moment starting from frame x_i to x_j delivers the same semantic meaning as the query S .

3.2 Language Representation via Sequential Embedding

We first extract the feature of an input query sentence. For each word s_i in the input sentence S , we generate its embedding vector $\mathbf{w}_i \in \mathbb{R}^{d^S}$ by the GloVe word2vec model [43], where d^S is the vector length. We then sequentially feed the word embeddings $\{\mathbf{w}_i\}_{i=0}^{l^S-1}$ into a three-layer bidirectional LSTM network [44], and use its average output as the feature representation of the input sentence, *i.e.* $\mathbf{f}^S \in \mathbb{R}^{d^S}$. The extracted feature encodes the language structure of the query sentence, thus describes the moment of interest.

3.3 Video Representation via 2D Temporal Feature Map

In this section, we extract moment features from the input video stream and then construct a 2D temporal feature map.

Given an input video, we first segment it into small non-overlapped video clips, where each clip consists of T contiguous frames. For each video clip, we extract its feature using pre-trained CNN model (see experiments section for details). To generate more compressed video clip representation in the channel dimension, we then feed the video clip feature into a fully connected layer with d^V output channels. Therefore, the final representation of compressed video clips is represented as $\{\mathbf{f}^V\}_{i=0}^{N-1}$, where d^V is the number of output channels and N is the total number of video clips.

The N video clips serve as the basic elements for moment candidate construction. Thus, we build up feature maps of moment candidates by the video clip features $\{\mathbf{f}^V\}_{i=0}^{N-1}$. Previous works extract moment features from clip features in two ways: pooling [6] or stacked convolution [35]. In this work, we follow the stacked convolution design. In the stacked convolution, the output of each layer are moment features, where these moments have equal length but start at different time, as shown in Fig. 2. In order to generate moment features, a simple way is to stack N convolution layers, where each layer has a kernel of size 2 and stride 1 (except for the first layer, which has kernel size 1 and stride 1). However, when N is large, it is infeasible

to fit the data into memory. Instead of enumerating all of them, we conduct a sparse sampling strategy as following: We first define two constant number A and K . For the first convolution layer, we set both the kernel size and stride to 1. For the $(i+1)\frac{A}{2}$ -th layers ($1 \leq i \leq K-1$), the kernel size and stride are set to 3 and 2, respectively. All the rest layers have kernels of size 2 and strides 1. We also add batch norm after each convolution layer and use *Tanh* as the activation function. Therefore, there are $\frac{(K+1)A}{2}$ convolution layers in total. In this way, we densely sample moments of short duration, and gradually increase the sampling interval when the moment duration becomes long. In more details, when the number of sampled clips is small, *i.e.* $N \leq A$, we enumerate all possible moments as candidates. When N becomes large, *i.e.* $N > A$, a moment starting from clip v_a to v_b is selected as the candidate when satisfying the following condition $G(a, b)$:

$$G(a, b) \Leftarrow (a \bmod s=0) \ \& \ (b \bmod s=0), \quad (1)$$

where a and b are the indexes of clips, s is defined as $s = 2^{k-1}$, $k = \lceil \log_2(\frac{b-a+1}{A}) + 1 \rceil$ and $\lceil \cdot \rceil$ is the ceiling function. If $G(a, b) = 1$, the moment is selected as the candidate, otherwise, it is not selected. This sampling strategy can largely reduce the number of moment candidates, as well as the computational cost.

Different from previous methods which directly operate on an individual video moment, we restructure the whole sampled moments to a 2D temporal feature map, denoted as $\mathbf{F}^M \in \mathbb{R}^{N \times N \times d^V}$. The 2D temporal feature map \mathbf{F}^M consists of three dimensions: the first two dimensions N represent the start and end clip indexes respectively, while the third one d^V indicates the feature dimension. The feature of a moment starting from clip v_a and during b clips is located at $\mathbf{F}^M[a, b, :]$ on the feature map. Denoting the i -th output at j -th convolution layer as $\mathbf{f}^M[i, j] \in \mathbb{R}^{d^V}$, it corresponds to the (a, b) -th location on the feature map \mathbf{F}^M , where

$$\begin{aligned} a &= 2^k i, \\ b &= \begin{cases} j & \text{if } j < A, \\ A + 2^{\lceil \frac{2(j-A+1)}{A} \rceil} \cdot (j - A + 1) - 1 & \text{otherwise.} \end{cases} \end{aligned} \quad (2)$$

Noted that, the moment's start and end clip indexes a and b should satisfy $a + b < N$. Therefore, on the 2D temporal feature map, all the moment candidates locating at the region of $a + b \geq N$ are invalid, *i.e.* the lower triangular part of the map, as shown in Figure 2. The values in this region are padded with zeros in implementation.

3.4 Multi-Scale 2D Temporal Adjacent Network

After obtaining the language and video features, we then predict the best matching moment queried by the sentence from all candidates.

We first fuse the 2D temporal feature maps \mathbf{F}^M with the sentence feature \mathbf{f}^S . Specifically, we project these two cross-domain features to the same subspace via fully connected layers, and then fuse them through Hadamard product and ℓ_2 normalization as

$$\mathbf{F}^F = \|(\mathbf{w}^S \cdot \mathbf{f}^S \cdot \mathbf{1}^T) \odot (\mathbf{W}^M \cdot \mathbf{F}^M)\|_F, \quad (3)$$

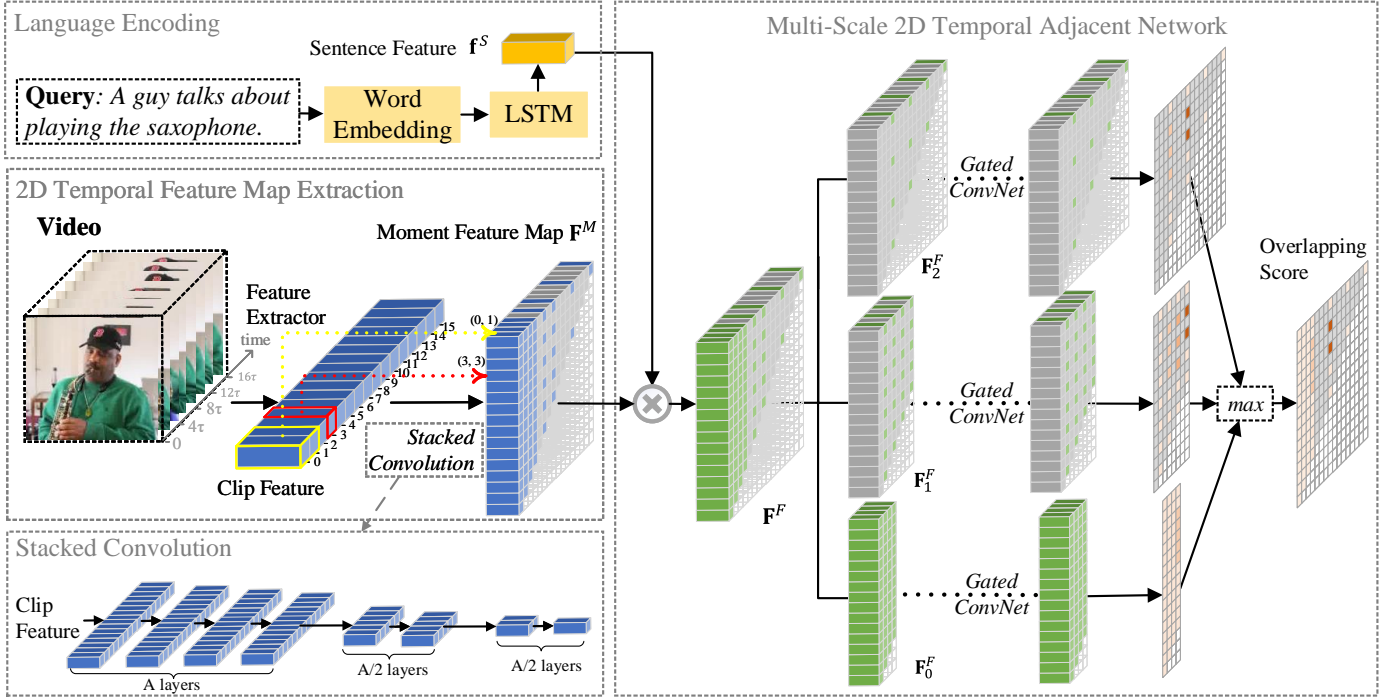


Fig. 2: Our proposed framework. The framework is composed of three modules: the language encoding module, the 2D temporal feature map extraction module, and the multi-scale 2D temporal adjacent network.

where \mathbf{w}^S and \mathbf{W}^M represent the learnt parameters of the fully connected layers, $\mathbb{1}^T$ is the transpose of an all-ones vector, \odot is Hadamard product, and $\|\cdot\|_F$ denotes Frobenius normalization. The fused feature map is denoted as $\mathbf{F}^F \in \mathbb{R}^{N \times N \times d^F}$, where d^F is the feature dimension after fusion.

Different from our preliminary work which directly builds up a temporal adjacent network over the single-scale map \mathbf{F}^F , in this paper, we build up multi-scale temporal adjacent networks over the sparse multi-scale maps. In more details, we first restructure the single-scale map \mathbf{F}^F to K multi-scale maps. The k -th sparse map is sampled from the fused feature map with interval 2^k . Specifically, all the $(2^k i - 1, 2^k j - 1)$ -th locations on \mathbf{F}^F are sampled, where $1 \leq i \leq N$ and $1 \leq j \leq A$ and A defines the number of anchors at each scale. In this way, the shortest and longest moments in the k -th map are of size 2^k and $2^k A$, respectively. We denote the multi-scale map as $\{\mathbf{F}_k^F | \mathbf{F}_k^F \in \mathbb{R}^{N \times 2^k A \times d^F}, 0 \leq k \leq K-1\}$.

Next, we build up temporal adjacent networks over the multi-scale 2D feature maps. In more details, each map corresponds to L gated convolutional layers [45] with kernel size of κ , dilation size of 2^k and stride size of 2^{k+1} . The output of the L layers keeps the same shape as the input fused feature map through zero padding at each layer. This design enables the model to gradually perceive more context of adjacent moment candidates, while learn the difference between moment candidates. Moreover, the receptive field of the network is large, thus it can observe a large portion of

video content, resulting in learning the temporal contexts.

Finally, we predict the matching scores of moment candidates with the given sentence on the multi-scale 2D temporal maps. We feed the output feature maps into fully connected layers and the *sigmoid* function separately, then generate multi-scale 2D score maps. The valid scores on each score map are then collected, denoted as $P_k = \{p_k^i\}_{i=1}^{C_k}$, where C_k is the total number of valid moment candidates of the k -th map. Each value p_k^i on the k -th map represents the matching score of a moment candidate with the sentence.

3.5 Training and Inference

During training, we adopt a scaled *IoU* value as the supervision signal, rather than a hard binary score. Specifically, for each moment candidate, we compute its *IoU* o_k^i with the ground truth moment. The *IoU* score o_k^i is then scaled with thresholds 0.5 as

$$y_k^i = \begin{cases} 0 & o_k^i \leq 0.5, \\ 2o_k^i - 1 & o_k^i > 0.5, \end{cases} \quad (4)$$

and y_k^i serves as the supervision label. Our network is trained by a binary cross entropy loss as

$$Loss = \sum_{k=0}^{K-1} \sum_{i=1}^{C_k} y_k^i \log p_k^i + (1 - y_k^i) \log(1 - p_k^i), \quad (5)$$

where K is the total number of maps, p_k^i and C_k are the output score of a moment and the total number of valid candidates on the k -th map.

During inference, the score maps $\{P_k, 0 \leq k \leq K-1\}$ are recovered to a single-scale map P' based on the moment location at the original single-scale map. Noted that some moments are predicted by more than one score map. And we choose the highest score as its final prediction.

1. Since the dilation and stride is equal, during our implementation, we reformat the k -th sparse maps ($k > 0$) to dense maps with stride 2^k and do convolution operation with stride 1 and dilation 1. This is equal to the previous, however, with less memory cost.

4 EXPERIMENTS

In this section, we first introduce the datasets used in our experiments and then go through the experiment settings. Next, we compare our proposed MS-2D-TAN with previous work. Finally, we conduct ablation studies and visualize experimental results.

4.1 Datasets

We evaluate our methods on the following three datasets:

Charades-STA [5]. It contains 9,848 videos of daily indoor activities. It is originally designed for action recognition and localization. Gao *et al.* [5] extend the temporal annotation (*i.e.* the start and end time of moments) of this dataset with language descriptions. Charades-STA contains 12,408 moment-sentence pairs in training set and 3,720 pairs in testing set.

ActivityNet Captions [46]. It consists of 19,209 videos, whose content are diverse and open. It is originally designed for dense video captioning, and recently introduced to moment localization with natural language, since these two tasks are reversible [16], [17]. Following the experimental setting in [17], we use val_1 as the validation set and val_2 as the testing set, which have 37,417, 17,505, and 17,031 moment-sentence pairs for training, validation, and testing, respectively. Currently, this is the largest dataset in this task.

TACoS [47]. It consists of 127 videos selected from the MPII Cooking Composite Activities video corpus [48], which contain different activities happened in the kitchen room. Regneri *et al.* extend the sentence descriptions by crowd-sourcing. A standard split [5] consists of 10,146, 4,589, and 4,083 moment-sentence pairs for training, validation and testing, respectively.

We also demonstrate the distribution of video duration, target moment duration and their ratio of these three datasets in Fig. 3. We can observe that videos in TACoS dataset have more variant and longer duration than others (see Fig. 3 Left). Charades-STA has shorter target moments compared to ActivityNet Captions and TACoS (see 3 Middle). By comparing the ratio between the target moment and the entire video (see Fig. 3 Right), we find that most target moments in TACoS have smaller ratios.

4.2 Evaluation Metric

Following the setting in previous work [5], we evaluate our model by computing *Rank n@m*. It is defined as the percentage of language queries having at least one correct moment retrieval in the top-*n* retrieved moments. A retrieved moment is correct when its IoU with the ground truth moment is larger than *m*. There are specific settings of *n* and *m* for different datasets. Specifically, we report the results as $n \in \{1, 5\}$ with $m \in \{0.5, 0.7\}$ for Charades-STA dataset, $n \in \{1, 5\}$ with $m \in \{0.3, 0.5, 0.7\}$ for ActivityNet Captions dataset, and $n \in \{1, 5\}$ with $m \in \{0.1, 0.3, 0.5, 0.7\}$ for TACoS dataset.

4.3 Implementation Details

The performance of video features plays an important role in moment localization with natural language. To make a

TABLE 1: Network settings of different datasets.

	Charades-STA	ActivityNet Captions	TACoS
<i>N</i>	64	128	512
<i>A</i>	16	16	8
<i>L</i>	2	4	2
<i>K</i>	17	9	9
<i>S</i>	3	4	5

thorough and fair comparison, we collect the most commonly used features in previous works, including VGG [49], C3D [50] and I3D [51] features. The details about video feature extractor are illustrated as follows:

- **VGG**. Following [35], we use VGG16 pre-trained on ImageNet [52]. Specifically, videos are decoded at 24 fps and the output of *fc7* layer after *ReLU* are extracted at 6 fps. The clip feature corresponds to every 4 consecutive frame. Therefore, each clip corresponds to 1 second.
- **C3D**. Following [38], we use C3D network pre-trained on sport1M [53]. Specifically, videos are decoded at 16 fps and the output of *fc6* layer after *ReLU* are extracted for every 16 consecutive frames. Each video clip corresponds to 1 second.
- **I3D**. Following [18], we use I3D network pre-trained on Kinetics [51]. Specifically, videos are decoded at 25 fps and the output of the last average pooling layer are extracted for every 16 consecutive frames. Therefore, each video clip corresponds to 0.64 second. For Charades-STA, we also finetune the I3D on the Charades dataset [54].

Some predicted moments highly overlap with each other. To reduce the redundant prediction, we adopt non-maximum suppression (NMS) based on the predicted scores P' . We fix the IoU threshold for NMS to 0.49 for all the experiments. After NMS, we use the top-*n* ranked moments for evaluation. Please note that NMS does not affect the top-1 result.

During training, we adopt sliding window to random select *N* consecutive clips. On Charades-STA and TACoS, we set *N* to 64 and 512 respectively. For ActivityNet Captions, we rescale each feature sequence to 128 and set *N* to 128, following [24]. We also list the hyper parameters with the best performance for the three datasets, as shown in Table 1. All these anchors and scale settings can cover at least 98% of training moments with an IoU threshold of 0.7. To train MS-2D-TAN from scratch, the learning rate is set to 0.0001 without weight decay. The batch size is set to 32 and Adam [55] is used as optimizer. The hidden state sizes of all layers in our model are set to 512.

4.4 Comparison with State-of-the-Art Methods

We evaluate the proposed MS-2D-TAN approach on three benchmark datasets, and compare it with recently proposed state-of-the-art methods², including:

- **clip-based methods**:
 - **anchor based methods**: TGN [16], CMIN [17] and CBP [38], SCDM [18],
 - **anchor free methods**: ACRN [12], ROLE [23], SLTA [28], DEBUG [27], VSLNet [19], GDP [26],

2. In addition to the video features, SLTA [28], SM-RL [30] use extra object features extracted from Faster R-CNN [57].

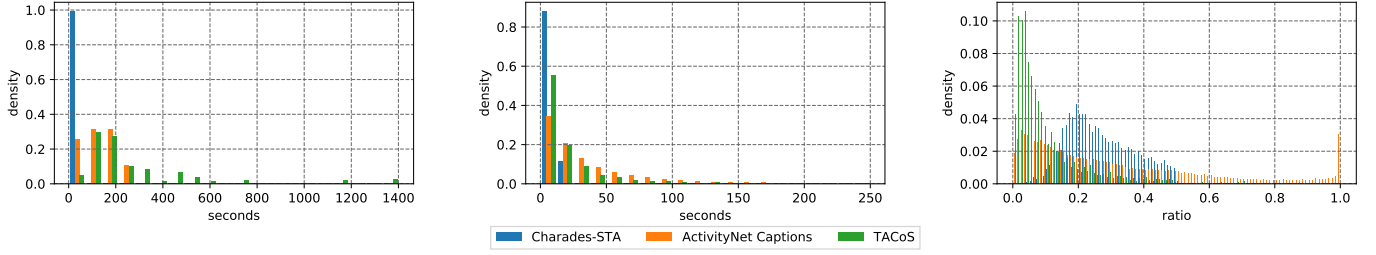


Fig. 3: Comparing Charades-STA, ActivityNet Captions and TACoS. **Left:** comparing video duration. **Middle:** comparing target moment duration. **Right:** comparing the ratio between target moment duration and video duration.

TABLE 2: Performance Comparison on Charades-STA.

Method	Rank1@		Rank5@	
	0.5	0.7	0.5	0.7
VGG features				
MCN [6]	17.46	8.01	48.22	26.73
MAN [35]	<i>41.24</i>	<i>20.54</i>	<i>83.21</i>	51.85
SM-RL [30]	24.36	11.17	61.25	32.08
SAP [34]	27.42	13.36	66.37	38.15
MS-2D-TAN	45.65	27.20	86.72	56.42
C3D features				
CTRL [5]	23.63	8.89	58.92	29.52
ACRN [12]	20.26	7.64	71.99	27.79
ROLE [23]	21.74	7.82	70.37	30.06
VAL [13]	23.12	9.16	61.26	27.98
ACL-K [11]	30.48	12.20	64.84	35.13
DEBUG [27]	37.39	17.69	—	—
GDP [26]	<i>39.47</i>	18.49	—	—
RWM-RL [29]	36.70	—	—	—
QSPN [33]	35.60	15.80	79.40	45.40
SLTA [28]	22.81	8.25	72.39	31.46
ABLR [20]	24.36	9.01	—	—
TripNet [31]	36.61	14.50	—	—
CBP [38]	36.80	<i>18.87</i>	70.94	50.19
TSP-PRL [32]	37.39	17.69	—	—
MS-2D-TAN	41.10	23.25	81.53	<i>48.55</i>
I3D features				
SCDM [18]	<i>54.44</i>	<i>33.43</i>	74.43	<i>58.08</i>
DRN [22]	53.09	31.75	<i>89.06</i>	<i>60.05</i>
MS-2D-TAN	56.64	36.21	89.14	61.13
I3D features finetuned on Charades				
ExCL [21]	44.10	22.40	—	—
TMLGA [25]	52.02	33.74	—	—
LGI [24]	<i>59.46</i>	<i>35.48</i>	—	—
VSLNet [19]	54.19	35.22	—	—
MS-2D-TAN	60.08	37.39	89.06	59.17

LGI [24], ABLR [20], TMLGA [25], ExCL [21] and DRN [22],

- **reinforcement learning based methods:** RWM-RL [29], SM-RL [30], TripNet [31] and TSP-RPL [32],
- **moment-based methods:**
 - **global/local context methods:** MCN [6], CTRL [5], ACL-K [11] and VAL [13],
 - **graph based methods:** MAN [35],
 - **cascade refinement methods:** QSPN [33] and SAP [34].

The results are summarized in Table 2–4. The values highlighted by **bold** and *italic* fonts indicate the top-2 methods, respectively. All results are reported in percentage (%).

The results show that MS-2D-TAN performs the best in various scenarios on all three benchmark datasets across different criteria. In most cases, MS-2D-TAN ranks the first

TABLE 3: Performance comparison on ActivityNet Captions.

Method	Rank1@			Rank5@		
	0.3	0.5	0.7	0.3	0.5	0.7
C3D features						
MCN [6]	39.35	21.36	6.43	68.12	53.23	29.70
CTRL [5]	47.43	29.01	10.34	75.32	59.17	37.54
TGN [16]	43.81	27.93	—	54.56	44.20	—
ACRN [12]	49.70	31.67	11.25	76.50	60.34	38.57
DEBUG [27]	55.91	39.72	—	—	—	—
GDP [26]	56.17	39.27	—	—	—	—
CMin [17]	63.61	43.40	23.88	<i>80.54</i>	67.95	<i>50.73</i>
RWM-RL [29]	—	36.90	—	—	—	—
QSPN [33]	52.13	33.26	13.43	77.72	62.39	40.78
ABLR [20]	55.67	36.79	—	—	—	—
TripNet [31]	48.42	32.19	13.93	—	—	—
SCDM [18]	54.80	36.75	19.86	77.29	64.99	41.53
CBP [38]	54.30	35.76	17.80	77.63	65.89	46.20
TSP-PRL [32]	56.08	38.76	—	—	—	—
DRN [22]	—	<i>45.45</i>	<i>24.36</i>	—	<i>77.97</i>	50.30
LGI [24]	58.52	41.51	23.07	—	—	—
MS-2D-TAN	<i>61.44</i>	46.93	28.23	86.67	78.79	61.29
I3D features						
ExCL [21]	<i>62.30</i>	42.70	24.10	—	—	—
TMLGA [25]	51.28	33.04	19.26	—	—	—
VSLNet [19]	63.16	<i>43.22</i>	<i>26.16</i>	—	—	—
MS-2D-TAN	61.16	46.56	29.92	86.91	78.02	60.10

or the second place³. It is worth noting that on TACoS dataset (see Table 4), our MS-2D-TAN surpasses the previous best approach CBP [38], by approximate 18 points and 25 points in term of Rank1@0.3 and Rank5@0.3, respectively. Moreover, on the large-scale ActivityNet Captions dataset, MS-2D-TAN also outperforms the top ranked method DRN [22] and VSLNet [19] with respect to IoU@0.5 and 0.7. It validates that MS-2D-TAN is able to localize the moment boundary more precisely.

In more details, by comparing MS-2D-TAN with other related methods, we obtain several observations. First, we compare MS-2D-TAN with previous moment-based methods: MCN [6], CTRL [5], ACL-K [11], VAL [13] and MAN [35]. From the results in Table 2–4, we observe that our MS-2D-TAN achieves superior results than concatenating local/global context. The reason is that independently matching the sentence with moment candidates ignores the temporal contexts, and cannot distinguish the small

3. Comparing CBP on Rank5@0.7 in Table 2, our MS-2D-TAN is worse due to the the IoU threshold for NMS. If the threshold is to set 0.5, the Rank5@0.5 and Rank5@0.7 of our model are 77.58 and 53.04, which outperform CBP.

TABLE 4: Performance comparison on TACoS.

Method	Rank1@				Rank5@			
	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7
VGG features								
MCN [6]	14.42	—	5.58	—	37.35	—	10.33	—
SM-RL [30]	26.51	20.25	15.95	—	50.01	38.47	27.84	—
SAP [34]	31.15	—	18.24	—	53.51	—	28.11	—
MS-2D-TAN	52.89	44.76	35.12	22.77	79.63	68.38	55.61	35.77
C3D features								
CTRL [5]	24.32	18.32	13.30	6.96	48.73	36.69	25.42	15.33
MCF [56]	25.84	18.64	12.53	—	52.96	37.13	24.73	—
TGN [16]	41.87	21.77	18.9	11.88	53.40	39.06	31.02	15.26
ACRN [12]	24.22	19.52	14.62	—	47.42	34.97	24.88	—
ROLE [23]	20.37	15.38	9.94	—	45.45	31.17	20.13	—
VAL [13]	25.74	19.76	14.74	—	51.87	38.55	26.52	—
ACL-K [11]	31.64	24.17	20.01	—	57.85	42.15	30.66	—
DEBUG [27]	41.15	23.45	—	—	—	—	—	—
GDP [26]	39.68	24.14	—	—	—	—	—	—
CMIN [17]	32.48	24.64	18.05	—	62.13	38.46	27.02	—
QSPN [33]	25.31	20.15	15.23	—	53.21	36.72	25.30	—
SLTA [28]	23.13	17.07	11.92	—	46.52	32.90	20.86	—
ABLR [20]	34.70	19.50	9.40	—	—	—	—	—
TripNet [31]	—	23.95	19.17	9.52	—	—	—	—
SCDM [18]	—	26.11	21.17	—	—	40.16	32.18	—
CBP [38]	—	27.31	24.79	19.10	—	43.64	37.40	25.59
DRN [22]	—	—	23.17	—	—	—	33.36	—
MS-2D-TAN	52.39	45.61	35.77	23.44	79.26	69.11	57.31	36.09
I3D features								
ExCL [21]	—	45.50	28.00	13.80	—	—	—	—
TMLGA [25]	—	24.54	21.65	16.46	—	—	—	—
VSLNet [19]	—	29.61	24.27	20.03	—	—	—	—
MS-2D-TAN	53.24	45.96	36.59	24.79	78.73	68.53	57.99	37.94

differences between the overlapped moments. Differently, our proposed MS-2D-TAN models the relations between moment candidates by a series of sparse 2D temporal maps, and enables the network to perceive more context information from the adjacent moment candidates. Hence, it gains large improvements compared to the methods that only consider global or local context. The closest work, MAN [35], utilizes GCN to model the relations among the moments and outperforms its previous methods on Charades-STA with VGG features. The experiment results demonstrate the effectiveness of the moment-level relation modeling. However, the nodes in GCN are permutation invariant, thus they ignore the temporal ordering of different moments. In contrast, our MS-2D-TAN models the temporal relations through a 2D convolution network and it can naturally learn the temporal ordering through its convolution kernel. From Table 2, we can observe that MS-2D-TAN surpasses MAN in all evaluation metrics.

Moreover, we compare our approach with clip-based methods (including anchor-based methods, anchor-free methods and RL-based methods). Previous anchor-based methods include TGN [16], CMIN [17], CBP [38] and SCDM [18]. Due to the involvement of more context information during prediction, the anchor-based approaches perform better than the global/local context approaches, however, inferior to our proposed MS-2D-TAN method. Anchor-based approaches implicitly learn the moment context information through a recurrent memory module or a 1D convolution network, while our MS-2D-TAN explicitly exploits the long range context information via the sparse multi-scale 2D temporal map. It further verifies the

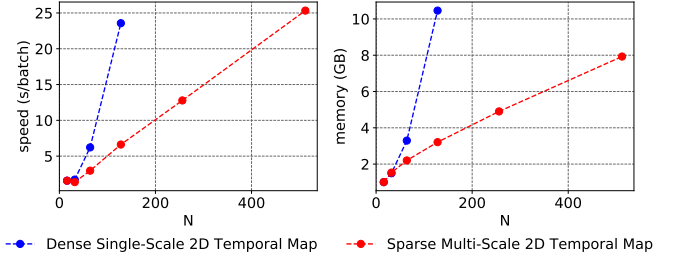


Fig. 4: Comparison on the inference speed and memory cost. N is number of video clips. **Left:** inference speed. **Right:** memory cost.

effectiveness of our model in high quality moment localization. We also compare our method with the anchor-free methods [12], [19]–[28], reinforcement learning based methods [29]–[32] and cascade refinement methods [33], [34]. Noted that anchor free methods and reinforcement learning methods do not rely on predefined anchors and directly predict the start and end time stamp. Therefore, their upper bounds under any IoU values should all be 100%, which is not directly comparable with our methods. Even with lower upper bound, our model still achieves superior performance compared to theirs. Compared with cascade refinement methods, our method predicts moments in one stage. While it is simple in design, it still outperforms these two methods. This demonstrates the significance of introducing moment context information. We also find that the performance improvement in TACoS dataset is larger than the other two datasets. This also verifies that our MS-2D-TAN is more efficient in modeling long videos.

4.5 Ablation Study

We conduct all our experiments with C3D features in our ablation study. We have shown the best hyper parameter settings of all three datasets in Table 1. In this section, we first verify the efficiency of our MS-2D-TAN from both theoretical proof and experiments. We then verify the effectiveness of the sparse multi-scale map and gated convolution of our model in Table 5. Next, we conduct several ablation studies on the hyper parameters of our model in Table 6, *i.e.* the kernel and layer setting, the number of anchors, window size and the number of hidden states. We also compare the number of candidate moments with previous methods.

4.5.1 Memory Usage and Speed

In this section, we compare the speed and memory cost between the dense single-scale map and sparse multi-scale map. For dense single-scale map, the total number of candidate moments is $\sum_{i=1}^N i = O(N^2)$. However, in the sparse multi-scale map, the total number of candidate moments is

$$\begin{aligned}
 & A \cdot N + A \cdot \frac{N}{2} + \dots + A \cdot \frac{N}{2^{K-1}} \\
 &= (2 - \frac{1}{2^{K-1}})AN \\
 &= O(N)
 \end{aligned} \tag{6}$$

In most cases, the number of video clips N is much larger than the number of anchors A and the number of maps K .

TABLE 5: Ablation study on the convolution layer and the 2D temporal map. SS and MS represent the dense single-scale map and the sparse multi-scale map. C, G represent MS-2D-TAN with convolution layers and gated convolution layers.

Row#	Input	Conv	Charades-STA				ActivityNet Captions						TACoS							
			Rank1@		Rank5@		Rank1@			Rank5@			Rank1@				Rank5@			
			0.5	0.7	0.5	0.7	0.3	0.5	0.7	0.3	0.5	0.7	0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7
1	SS	C	39.49	20.83	80.59	50.11	61.61	45.73	27.98	85.30	76.28	62.17	39.57	30.74	21.97	14.02	68.08	55.24	41.49	24.72
2	MS	C	<i>40.97</i>	<i>22.55</i>	<i>80.91</i>	46.48	60.22	<i>46.52</i>	29.98	<i>86.25</i>	<i>78.29</i>	<i>61.19</i>	<i>51.64</i>	<i>44.61</i>	<i>35.29</i>	<i>23.09</i>	<i>79.23</i>	<i>68.63</i>	<i>56.04</i>	<i>35.57</i>
3	MS	G	41.10	23.25	81.53	<i>48.55</i>	<i>61.44</i>	46.93	<i>28.23</i>	86.67	78.79	<i>61.29</i>	52.39	45.61	35.77	23.44	79.26	69.11	57.31	36.09

TABLE 6: Ablation study on hyper parameters. H , N , and S represent the size of each hidden layer, sliding window size, and the number of scales. A , K , and L represent the number of anchors, kernel size, and the number of layers of gated convolutions at each scale, respectively.

Row#	Method	Hyper Parameters							Rank1@				Rank5@				#Params ($\times 10^6$)
		H	N	S	A	κ	L		0.1	0.3	0.5	0.7	0.1	0.3	0.5	0.7	
1	Upper Bound	—	—	1	128	—	—	—	100.00	99.40	98.95	96.38	100.00	99.40	98.95	96.38	—
2		—	—	4	16	—	—	—	100.00	99.40	98.95	96.38	100.00	99.40	98.95	96.38	—
3		—	—	1	8	—	—	—	91.90	68.53	54.16	41.61	91.90	68.53	54.16	41.61	—
4		—	—	2	8	—	—	—	96.68	85.80	72.81	64.31	96.68	85.80	72.81	64.31	—
5		—	—	3	8	—	—	—	99.23	94.65	89.35	80.93	99.23	94.65	89.35	80.93	—
6		—	—	4	8	—	—	—	100.00	98.10	95.65	92.58	100.00	98.10	95.65	92.58	—
7		—	—	5	8	—	—	—	100.00	99.40	98.95	96.38	100.00	99.40	98.95	96.38	—
8		—	—	6	8	—	—	—	100.00	100.00	99.75	98.20	100.00	100.00	99.75	98.20	—
9		—	—	7	8	—	—	—	100.00	100.00	100.00	98.90	100.00	100.00	100.00	98.90	—
10		—	—	6	4	—	—	—	100.00	99.40	98.95	96.35	100.00	99.40	98.95	96.35	—
11	MS-2D-TAN	512	512	4	16	9	2	—	47.79	39.74	30.64	20.32	76.33	64.53	52.11	33.57	19.87
12		512	512	6	4	9	1	—	45.04	37.24	27.54	16.72	69.26	58.84	48.26	33.84	14.30
13		512	512	5	8	1	1	—	21.04	12.87	6.17	1.87	55.59	39.02	21.94	9.32	21.64
14		512	512	5	8	9	1	—	49.81	41.71	33.07	20.89	76.03	66.63	55.39	36.22	12.67
15		512	512	5	8	5	4	—	49.54	42.59	33.49	23.64	76.03	64.68	53.16	36.64	15.17
16		512	512	5	8	17	1	—	50.61	43.14	35.17	24.32	76.31	66.33	56.54	39.79	39.99
17		512	512	1	8	9	2	—	40.44	32.52	25.49	17.70	59.39	45.69	35.14	23.84	53.42
18		512	512	2	8	9	2	—	43.14	36.84	30.42	22.44	67.78	55.49	44.26	32.37	98.33
19		512	512	3	8	9	2	—	48.81	42.29	34.09	23.64	73.21	63.01	52.39	34.99	14.33
21		512	512	5	8	9	2	—	52.39	45.61	35.77	23.44	79.26	69.11	57.31	36.09	23.31
22		512	512	6	8	9	2	—	51.14	42.69	32.64	21.04	77.93	65.93	54.06	36.52	27.80
23		512	512	7	8	9	2	—	49.86	42.24	32.97	21.47	80.03	67.26	54.79	35.02	32.29
24		512	256	5	8	9	2	—	50.69	43.96	35.34	24.34	77.61	66.36	54.81	37.07	23.31
25		512	128	5	8	9	2	—	48.56	41.64	33.34	22.19	76.36	65.83	53.79	35.62	23.31
26		128	512	3	8	9	2	—	44.64	38.24	31.54	22.32	69.81	58.96	49.54	33.87	9.52
27		162	512	3	8	9	2	—	44.21	38.72	31.22	22.44	72.26	62.88	51.81	35.49	14.99
28	CBP	—	—	1	32	—	—	—	<i>27.31</i>	<i>24.79</i>	<i>19.10</i>	—	<i>43.64</i>	<i>37.40</i>	<i>25.59</i>	—	15.34

We also measure the computational cost through experiments, as shown in Fig. 4. The memory usage of sparse multi-scale map scales linearly with the sequence length, while dense single-scale map increases quadratically as the sequences become longer⁴. From both the theoretical and the experimental results, we find that the sparse multi-scale modeling has faster speed and lower memory cost compared to the dense single-scale modeling.

4.5.2 Effectiveness of Sparse Multi-Scale Map

In this section, we verify the effectiveness of our sparse multi-scale map, as shown in Table 5. Comparing row 1 and row 2, we find that even with less number of moment candidates in our sparse multi-scale map (row 2), we can still obtain better results compared to the dense single-scale map (row 1) in general. Among the three datasets, the largest improvement comes from TACoS dataset. The reason is that the videos in TACoS have longer average duration than the other datasets, as shown in Fig. 3. The dense single-scale map cannot handle long videos well, since the

receptive field is limited and context information from long range is missed. In contrast, the sparse multi-scale map has larger receptive field and involves more long range context and therefore outperforms the previous in TACoS dataset.

4.5.3 Effectiveness of Gated Convolution

In this section, we verify the effectiveness of the gated convolution in Table 5. Comparing row 2 and row 3, we find applying an additional gate on the convolution can improve the performance on TACoS dataset. The better performance of gated convolution is also consistent with its effectiveness in image inpainting [45]. Since videos in TACoS dataset are longer, more multi-scale maps are involved compared to the other two datasets (see Tab. 1). The gated convolution is more flexible to adjust the weights of moments among different scales compared to the conventional convolution.

4.5.4 Number of Scales

In this section, we evaluate the performance under different number of scales S . We vary the number of scales S from 1 to 7 in our MS-2D-TAN model. The results are shown in Table 5 (row 17 – 23). We observe that, increasing S from 1

4. Since the CuDNN package optimizes memory usage at run time, we turn it off for this comparison.

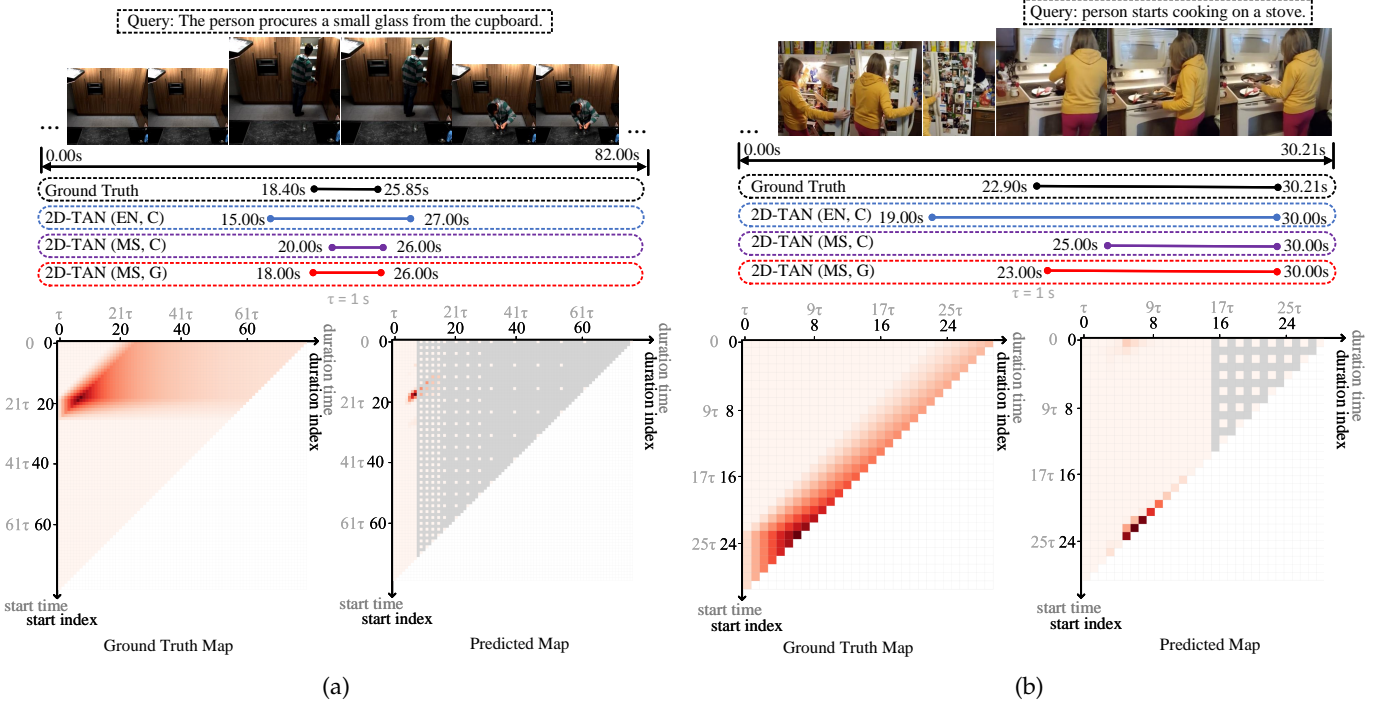


Fig. 5: Prediction examples of our model and the baselines. The predicted map is the result of our full model.

to 5 brings improvements (25.49 *v.s.* 30.42 *v.s.* 34.09 *v.s.* 35.04 *v.s.* 35.77 in *Rank5@0.5*). This observation is also consistent with the theoretical upper bound, as listed in Table 5 (Row 3 – 7). Here, the upper bound represents the performance of an ideal model that can provide a correct prediction on all the sampled video clips. The upper bound is smaller than 100% since the sampling of video moments may not cover all possible results. We also observe that further increasing the scale S from 5 to 7 does not benefit the overall performance. Since the upper bounds (Row 7–9) are already saturated, further increasing moment candidates may introduce additional data imbalance problem.

4.5.5 Number of Anchors

In this section, we evaluate the performance under different number of anchors A . As mentioned in Sec. 4.5.4, the upper bounds are vital to the overall performance. However, it is not the only factor. With the same upper bounds (row 2, 7, 10), we also vary the anchor size and the number of scales, as shown in Table 5 (row 11, 12, 14 and 21). From the experiments, we find that using $S = 5$ and $A = 8$ (row 21) achieves the best performance in the TACoS evaluation metrics. Using more maps with fewer anchors for each map may not provide sufficient context information at each scale (row 12 *v.s.* row 14). Meanwhile, using less maps with more anchors at each map involves less long range context at each scale due to the smaller receptive field. (row 11 *v.s.* row 21).

4.5.6 Kernel and Layer Setting

In this section, we evaluate different kernel size κ and number of layers L . With the same number of scales ($S = 5$), we vary the kernel size and layers at each scale, as shown in Table 5 (row 13 – 16 and row 21). There are several observations. First, we compare the model with $K = 1, L = 1$ (row

13) and $K = 9, L = 1$ (row 14). Row 13 predicts scores for each moment independently and row 14 jointly considers other neighboring moments in the 2D temporal map. The significant improvement from these two rows verify the importance of modeling context information from adjacent moments. Further enlarging the receptive field can achieve better performance (row 15, 16, and 21). Comparing row 15, 16 and 21 with the same receptive field size, we observe that the performance benefits more from larger kernel with fewer layers (row 16 and row 21) compared to small kernel with more layers (row 15).

4.5.7 Effectiveness of Window Size

In this section, we compare the performance with different sliding window sizes (N) during training in Table 5 (row 24, 25 and 21). We observe that larger window size can benefit the overall performance, since more context moments are observed during training. By using 512 clips extracted by C3D features (row 21), we can cover approximate 96.42% video length in average. And therefore it achieves the best performance compared with the row 25 (covering 87.57%) and row 24 (covering 67.47%).

4.5.8 Effectiveness of Hidden State Size

In this section, we evaluate the performance under different hidden state size H , as shown in (Row 26, 27, and 21). We observe that with larger number of hidden size, our model can achieve better performance. Even with about $\frac{2}{3}$ parameters of the previous best approach CBP [38], our model can achieve superior performance (Row 27 *v.s.* 28).

4.5.9 Number of Candidate Moments

Proposal based methods can benefit from a larger number of moments. In this section, we verify whether the superior

performance of our proposed MS-2D-TAN comes from the larger number of moments. The number of moment candidates is a vital factor in moment localization models. We first tune this factor in our MS-2D-TAN approach, and show its impacts on final performance. Next, we compare the previous best approach CBP [38] with respect to this factor. The upper bound performance of proposal based methods is decided by the selection of moment candidates. Therefore, we evaluate our MS-2D-TAN with the same moment selection of previous state-of-the-art proposal based methods. For a fair comparison, we compare CBP with our MS-2D-TAN on TACoS. In CBP (Row 27), each clip corresponds to 32 anchors, while in the MS-2D-TAN with $A = 8$ and $S = 3$ (Row 26), the corresponding anchors are a subset of the CBP anchors (16 anchors). With approximately the same number of parameters and half less number of anchors, our model outperforms the previous best approach by a large margin.

4.6 Qualitative Analysis

We provide some qualitative examples to validate the effectiveness of our MS-2D-TAN. As shown in Fig 5, the sparse multi-scale map exploits richer context information with a larger receptive field compared to the dense single-scale map. By combining with gated convolution, the location of the desired moment can be more precise.

We also visualize the predicted scores of our model. Comparing with the ground truth map, we can observe that the predicted map can correctly localize the target moment, while the surrounding moments also have high predicted scores. This demonstrates that our model can learn discriminative features from the neighboring moments.

5 CONCLUSION

In this paper, we study the problem of moment localization with natural language and present a novel Multi-Scale 2D Temporal Adjacent Networks (MS-2D-TAN) method. The core idea is to retrieve a moment on the multi-scale two-dimensional temporal map, which considers adjacent moment candidates as the temporal context. MS-2D-TAN is capable of encoding adjacent temporal context, while learning discriminative feature for matching video moments with the sentence query. Our model is simple in design and achieves competitive performance in comparison with other state-of-the-art methods on three benchmark datasets.

6 ACKNOWLEDGEMENT

This work is supported in part by NSF awards IIS-1704337, IIS-1722847, and IIS-1813709, as well as our corporate sponsors.

REFERENCES

- [1] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *ICCV*, 2017.
- [2] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *CVPR*, 2016.
- [3] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing web videos using titles," in *CVPR*, 2015.
- [4] W.-S. Chu, Y. Song, and A. Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," in *CVPR*, 2015.
- [5] J. Gao, C. Sun, Z. Yang, and R. Nevatia, "TALL: Temporal activity localization via language query," in *ICCV*, 2017.
- [6] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with natural language," in *ICCV*, 2017.
- [7] J. Lei, L. Yu, M. Bansal, and T. L. Berg, "TVQA: Localized, compositional video question answering," in *EMNLP*, 2018.
- [8] D. Shao, Y. Xiong, Y. Zhao, Q. Huang, Y. Qiao, and D. Lin, "Find and focus: Retrieve and localize video events with natural language queries," in *ECCV*, 2018.
- [9] X. Duan, W. Huang, C. Gan, J. Wang, W. Zhu, and J. Huang, "Weakly supervised dense event captioning in videos," in *NIPS*, 2018.
- [10] S. Gella, M. Lewis, and M. Rohrbach, "A dataset for telling the stories of social media videos," in *EMNLP*, 2018.
- [11] R. Ge, J. Gao, K. Chen, and R. Nevatia, "MAC: Mining activity concepts for language-based temporal localization," in *WACV*, 2019.
- [12] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua, "Attentive moment retrieval in videos," in *SIGIR*, 2018.
- [13] X. Song and Y. Han, "VAL: Visual-attention action localizer," in *PCM*, 2018.
- [14] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2d temporal adjacent networks for moment localization with natural language," in *AAAI*, 2020.
- [15] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *ACM MM*. ACM, 2017.
- [16] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua, "Temporally grounding natural sentence in video," in *EMNLP*, 2018.
- [17] Z. Zhang, Z. Lin, Z. Zhao, and Z. Xiao, "Cross-modal interaction networks for query-based moment retrieval in videos," in *SIGIR*, 2019.
- [18] Y. Yuan, L. Ma, J. Wang, W. Liu, and W. Zhu, "Semantic conditioned dynamic modulation for temporal sentence grounding in videos," in *NeurIPS*, 2019.
- [19] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, "Span-based localizing network for natural language video localization," in *ACL*, 2020.
- [20] Y. Yuan, T. Mei, and W. Zhu, "To find where you talk: Temporal sentence localization in video with attention based location regression," in *AAAI*, 2019.
- [21] S. Ghosh, A. Agarwal, Z. Parekh, and A. Hauptmann, "ExCL: Extractive clip localization using natural language descriptions," in *NAACL*, 2019.
- [22] R. Zeng, H. Xu, W. Huang, P. Chen, M. Tan, and C. Gan, "Dense regression network for video grounding," *CVPR*, 2020.
- [23] M. Liu, X. Wang, L. Nie, Q. Tian, B. Chen, and T.-S. Chua, "Cross-modal moment localization in videos," in *ACM MM*, 2018.
- [24] J. Mun, M. Cho, and B. Han, "Local-global video-text interactions for temporal grounding," in *CVPR*, 2020.
- [25] C. Rodriguez, E. Marrese-Taylor, F. S. Saleh, H. Li, and S. Gould, "Proposal-free temporal moment localization of a natural-language query in video using guided attention," in *WACV*, March 2020.
- [26] L. Chen, C. Lu, S. Tang, J. Xiao, D. Zhang, C. Tan, and X. Li, "Rethinking the bottom-up framework for query-based video localization," in *AAAI*, 2020.
- [27] C. Lu, L. Chen, C. Tan, X. Li, and J. Xiao, "Debug: A dense bottom-up grounding approach for natural language video localization," in *EMNLP*, 2019.
- [28] B. Jiang, X. Huang, C. Yang, and J. Yuan, "Cross-modal video moment retrieval with spatial and language-temporal attention," in *ICMR*, 2019.
- [29] D. He, X. Zhao, J. Huang, F. Li, X. Liu, and S. Wen, "Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos," in *AAAI*, 2019.
- [30] W. Wang, Y. Huang, and L. Wang, "Language-driven temporal activity localization: A semantic matching reinforcement learning model," in *CVPR*, 2019.
- [31] M. Hahn, A. Kadav, J. M. Rehg, and H. P. Graf, "Tripping through time: Efficient localization of activities in videos," in *CVPR Workshop*, 2019.
- [32] J. Wu, G. Li, S. Liu, and L. Lin, "Tree-structured policy based progressive reinforcement learning for temporally language grounding in video," in *AAAI*, 2020.

- [33] H. Xu, K. He, B. A. Plummer, L. Sigal, S. Sclaroff, and K. Saenko, "Multilevel language and vision integration for text-to-clip retrieval," in *AAAI*, 2019.
- [34] S. Chen and Y.-G. Jiang, "Semantic proposal for activity localization in videos via sentence query," in *AAAI*, 2019.
- [35] D. Zhang, X. Dai, X. Wang, Y.-F. Wang, and L. S. Davis, "MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment," in *CVPR*, 2019.
- [36] N. Dehmamy, A.-L. Barabási, and R. Yu, "Understanding the representation power of graph neural networks in learning graph topology," in *NeurIPS*, 2019.
- [37] B. Liu, S. Yeung, E. Chou, D.-A. Huang, L. Fei-Fei, and J. Carlos Niebles, "Temporal modular networks for retrieving complex compositional activities in videos," in *ECCV*, 2018.
- [38] J. Wang, L. Ma, and W. Jiang, "Temporally grounding language queries in videos by contextual boundary-aware prediction," in *AAAI*, 2020.
- [39] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv:2004.05150*, 2020.
- [40] S. Sukhbaatar, E. Grave, P. Bojanowski, and A. Joulin, "Adaptive attention span in transformers," *ACL*, 2019.
- [41] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "BMN: Boundary-matching network for temporal action proposal generation," in *ICCV*, 2019.
- [42] C. Lin, J. Li, Y. Wang, Y. Tai, D. Luo, Z. Cui, C. Wang, J. Li, F. Huang, and R. Ji, "Fast learning of temporal action proposal via dense boundary generator," in *AAAI*, 2020.
- [43] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.
- [44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [45] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *ICCV*, 2019.
- [46] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, "Dense-captioning events in videos," in *ICCV*, 2017.
- [47] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal, "Grounding action descriptions in videos," *TACL*, 2013.
- [48] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, "Script data for attribute-based recognition of composite activities," in *ECCV*, 2012.
- [49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICPR*, 2015.
- [50] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.
- [51] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.
- [52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.
- [53] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.
- [54] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *ECCV*, 2016.
- [55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICPR*, 2015.
- [56] A. Wu and Y. Han, "Multi-modal circulant fusion for video-to-language and backward," in *IJCAI*, 2018.
- [57] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.



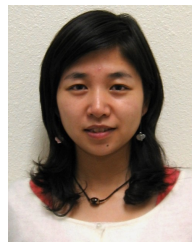
Songyang Zhang received the BS degree from Southeast University, China, in 2015 and MS degree from Zhejiang University, China, in 2018. He is currently working toward the PhD degree at University of Rochester. His research interests include computer vision and natural language processing.



tion, saliency detection, etc.



dows, Office, Bing Multimedia Search, Azure Media Service, Xiaolce, etc.



State Presidential Distinction Award and 2014 College Achievement Award. She also received the Best Paper award from ICME 2013 and ICIMCS 2012. She has obtained many competitive external grants from NSF, US Army, US Department of Defense and Texas Department of Transportation.



Jiebo Luo (S93, M96, SM99, F09) is a Professor of Computer Science at the University of Rochester which he joined in 2011 after a prolific career of fifteen years at Kodak Research Laboratories. He has authored nearly 500 technical papers and holds over 90 U.S. patents. His research interests include computer vision, NLP, machine learning, data mining, computational social science, and digital health. He has been involved in numerous technical conferences, including serving as a program co-chair of ACM Multimedia 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, as well as a general co-chair of ACM Multimedia 2018. He has served on the editorial boards of the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Multimedia (TMM), IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), IEEE Transactions on Big Data (TBD), ACM Transactions on Intelligent Systems and Technology (TIST), Pattern Recognition, Knowledge and Information Systems (KAIS), Machine Vision and Applications, and Journal of Electronic Imaging. He is the current Editor-in-Chief of the IEEE Transactions on Multimedia. Professor Luo is also a Fellow of ACM, AAAI, SPIE, and IAPR.