Proposal-free Temporal Moment Localization of a Natural-Language Query in Video using Guided Attention

Cristian Rodriguez Opazo[1,2] Edison Marrese-Taylor [3] Fatemeh Sadat Saleh[1,2]
Hongdong Li[1,2] Stephen Gould[1,2]

Australian National University [1], Australian Centre for Robotic Vision (ACRV) [2]
Graduate School of Engineering, The University of Tokyo [3]

Abstract

This paper studies the problem of temporal moment localization in a long untrimmed video using natural language as the query. Given an untrimmed video and a sentence as the query, the goal is to determine the starting, and the ending, of the relevant visual moment in the video, that corresponds to the query sentence. While previous works have tackled this task by a propose-and-rank approach, we introduce a more efficient, end-to-end trainable, and proposal-free approach *that relies on three key components: a dynamic filter to transfer language information to the visual domain, a new loss function to guide our model to attend the most relevant parts of the video, and soft labels to model annotation uncertainty. We evaluate our method on two benchmark datasets, Charades-STA and ActivityNet-Captions. Experimental results show that our approach outperforms state-of-the-art methods on both datasets.*

1. Introduction

Vision-and-language understanding is an important problem in computer vision, drawing increasing interest over the past few years, motivated by its vast potential applications. This setting includes tasks such as video captioning and video question answering. While promising results have been achieved in these tasks, a fundamental issue remains to be tackled, namely, that these informative video segments need to be manually trimmed (pre-segmented) and often aligned with the relevant textual descriptions that accompany them.

Since searching for a specific visual event over a long video sequence is difficult and inefficient to do manually, even for a small number of videos, automated search engines are needed to deal with this requirement, especially when dealing with a massive amount of video data. It is clear that these search engines have to retrieve videos not only based on the video metadata, but they also must exploit their internal information in order to accurately localize the required information/segment.

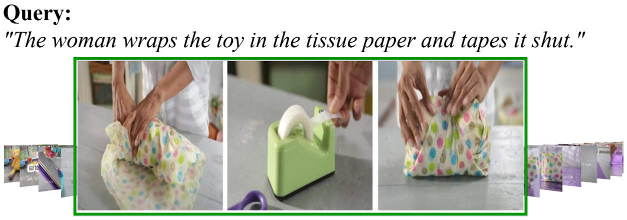
Query:
"The woman wraps the toy in the tissue paper and tapes it shut."

Figure 1: An illustration of temporal localization of a natural language query in an untrimmed video. Given a query and a video the task is temporally localize the starting and ending of the sentence in the video.

In light of this, automatically recognizing *when* an activity is happening in a video has become a crucial task in computer vision. Its applicability in other research areas such as video surveillance and robotics [32], among others, has also helped bring interest into this task. Earlier works in this area focused on *temporal action localization* [36, 29, 51, 55, 11, 5, 13], which attempted to localize "interesting" actions in a video from a predefined set of actions. However, this approach constrains the search engine to a relatively small and not realistic set of queries from users.

To address this issue the task of "temporal action localization with natural language" has been proposed recently [12, 19]. Given a query, the goal is to determine the start and end locations of the associated video segment in an untrimmed, long video. In this context, we are specifically interested in the problem of natural-language-based temporal localization, or temporal sentence localization in the video. Formally, given an untrimmed video and a query in natural language, the task is to identify the start and end points of the video segment in response to it, therefore effectively locating the temporal segment (i.e, moment) that best corresponds to the given query, as depicted in Figure 1.

Current approaches to the localization problem in computer vision, either spatial or temporal, mainly focus on creating a good multi-modal embedding space and generating proposals based on the given query. In these *propose and rank* approaches, candidate regions are first generated by a separate method and then fed to a classifier to get the probabilities of containing target classes, effectively ranking them. Despite the relative success of these approaches, this setting is ultimately restrictive in scope since it uses predefined clips as candidates, making it hard to extend for videos with considerable variance in length.

To this end, we propose an approach that does not rely on candidate generation or ranking, being able to directly predict the start and end times given a query in natural language. Our model is guided by a dynamic filter, which is in charge of matching the text and video embeddings, and a principal attention mechanism which encourages the model to focus on the features inside of segment of interest. To the best of our knowledge, our approach is the first to do so[1].

Recent works on temporal action localization with natural language [16] has adopted an approach akin to Machine Reading Comprehension (MC) [6], but in a multi-modal setting. Similar to ours, these models are trained in an end-to-end manner. Specifically, they maximize the likelihood of correctly predicting the start and end frames associated to a given query, in a way analogous to predicting the text span corresponding to the correct answer in MC. We note, however, that annotating the start and end of a given activity inside a video is a highly subjective task, as evidenced by the relative lower inter-annotator agreements scores [41, 1, 2]. In light of this, our model incorporates annotation subjectivity in a simple yet efficient manner, obtaining increased performance.

We conduct experiments on two challenging datasets, Charades-STA [12] and Activity Net Captions [26], demonstrating the effectiveness of our proposed method and obtaining state-of-the-art performance on both. Our results also empirically prove the effectiveness of our attention-based guidance mechanism, and of incorporating the subjective nature of the annotations into the model, ultimately validating our proposed approach through ablation analysis.

2. Related Work

2.1. Temporal Action Localization

The task of temporal action localization aims to solve the problem of recognizing and determining temporal boundaries of action instances in videos. Since activities (in the wild) consist of a diverse combination of actors, actions and objects over various periods of time, earlier work focused on classification of video clips that contained a single activity, i.e. where the videos were trimmed.

More recently there has also been significant work in localizing activities in longer, untrimmed videos. For example, Shou et al. [40] trained C3D [48] with a localization loss and achieved state-of-the-art performance on THUMOS [21]. On the other hand, Ma et al. [33] used a temporal LSTM to generate frame-wise prediction scores and then merged the detection intervals based on the predictions. Singh et al. [45] extended the two-stream [44] framework with person detection and bi-directional LSTMs and achieved state-of-the-art performance on the MPII-Cooking dataset [38].

Escorcia et al. [10] took a different approach and introduced an algorithm for generating temporal action proposals from long videos, which are used to retrieve temporal segments that are likely to contain actions. Lin et al. [30] proposed an approach based on 1D temporal convolutional layers to skip the proposal generation step via directly detecting action instances in untrimmed video.

The major limitation of these action localization methods is that they are restricted to a pre-defined list of actions. As it is non-trivial to design a label space which has enough coverage for such activities without losing useful details in users' queries this approach makes it difficult to cover complex activity queries.

2.2. Temporal language-driven moment localization

Language-driven temporal moment localization is the task of determining the start and end time of the temporal video segment that best corresponds to a given natural language query. Essentially, this means to use natural language queries to localize activities in untrimmed videos. While the language-based setting allows for having an open set of activities, it also corresponds to a more natural query specification, as it directly includes objects and their properties as well as relations between the involved entities.

The work of Hendricks et al. [19] and Gao et al. [12] are generally regarded as pioneer on this task. While Hendricks et al. [19] proposed to learn a shared embedding for both video temporal context features and natural language queries, suitable for matching candidate video clips and language queries using a ranking loss and handcrafted heuristics, Gao et al. [12] proposed to generate candidate clips using temporal sliding windows which are later ranked based on alignment or regression learning objectives.

The research line defined by [12], where proposals are generated using temporal sliding windows was later extended in [15], which leverage activity classifiers to help encode visual concepts, and add an *actionness score* to help capture the semantics from verb-object pairs in the queries. Recently, [31] also resorted to sliding windows for

[1] We have implemented our model using PyTorch. We plan to release our code upon acceptance

generating proposals, but used a memory attention model when matching each proposal to the input query. Despite their simplicity and ability to provide coarse control over the frames that are evaluated, the main problem with these methods is that the matching mechanism between the candidate proposals and the query is computationally expensive.

To tackle this issue some approaches have focused on reducing the number of temporal proposals generated. These methods generally focus on producing query-guided or query-dependent video clip proposals, skipping unlikely clips from the matching step and thus speeding up the whole process. In this context, [7] propose to capture frame-by-word interactions between video and language and then score a set of temporal candidates at multiple scales to localize the video segment that corresponds to the query. Similarly, [52] propose a multilevel model to tightly integrate language and vision features and then use a fine-grained similarity measure for query-proposal matching.

A slightly different but related approach is proposed by [20], where the video context is modeled as a latent variable to reason about the temporal relationships. The work of [54] further improved on this by utilizing a graph structured network to model temporal relationships among different moments, therefore addressing semantic and structural misalignment problems. On the other hand, [8] focused on the proposal generation step, integrating the semantic information of the natural language query into the proposal generation process to get discriminative activity proposals. Although previous methods use techniques to directly generate candidate moment representations aligned with language semantics instead of fetching video clips independently, they still depend on ranking a fixed number of temporal candidates in each video, leading to inefficiencies.

More recently, methods that go beyond the *scan and localize* approach, which can therefore directly output the temporal coordinates of the localized video segment have been proposed. For example, [53] used a co-attention based model for temporal sentence localization. In this context, some models also resort to reinforcement learning to dynamically observe a sequence of video frames conditioned on the given language query. Concretely, [50] train a recurrent neural network for language-driven temporal activity localization using this approach, while also utilizing Faster R-CNN [35] trained on the Visual Genome dataset [27] to obtain regional visual features and incorporate more semantic concepts to the model. Similarly, [17] use this approach and learn how to skip around the video, therefore being able to more easily localize relevant clips in long videos. Instead of simply concatenating the video representation and query embedding, their approach uses a gated attention architecture to model textual and visual representations in order to align the text and video content.

Finally, [16] proposes a simpler approach that does not rely on reinforcement learning and does not either involve retrieve and re-ranking multiple proposal segments. Their approach focuses on predicting the start and end frames by leveraging cross-modal interactions between the text and video. In this context, our method proposes a simple yet effective proposal-free approach which makes it more practical to use.

3. Proposed Approach

Let $V \in \mathcal{V}$ be a video that can be characterized as a sequence of frames such that $V = \{v_t\}$ with $t = 1, \dots, l$. Each video in $\mathcal{V}$ is annotated with a natural language passage $S \in \mathcal{S}$ where S is a sequence of words $S = \{s_j\}$ with $j = 1, \dots, m$, which describes what is happening in a certain period of time. Formally, this interval is defined by t^s and t^e, the starting and ending points of the annotations in time, respectively.

We propose a model that is trained end-to-end on a set of example tuples of annotated videos (V_k, S_k, t^s_k, t^e_k). Although in the data a given video may be annotated with more than one single moment, and one natural language description may be associated to multiple moments, in this work we assume each derived case as an independent, separate training example. Given a new video and sentence tuple (V_r, S_r), our model predicts the most likely temporal localization of the contents of S_r in terms of its start and end positions $t^{s\star}_r$ and $t^{s\star}_r$ in the video, therefore effectively solving the problem of temporal localization of sentences in videos. In the following, for simplicity we drop the index k associated to each training example.

Our model is designed in a modular way, offering more flexibility over previous work. There are four main components which we proceed to describe in the following sections. First, sections 3.1 and 3.2 give details about our video and natural language query encoders, respectively. These can be seen as the initial components in our model, responsible for effectively obtaining a semantically rich representation for the data coming from each input modality. The output representations returned by these modules are later combined using a dynamic filter layer, described in section 3.3, which allows us to transfer language information to the visual domain. Finally, section 3.4 describes our proposed localization layer, which takes the filtered video features and uses them to predict the start and end frames of the desired location. Figure 2 shows an overview of our proposed approach.

3.1. Video Encoder

As discussed earlier, previous works on temporal sentence localization in videos mostly rely on proposal generation, either using sliding windows or other heuristics [12, 19, 15, 31]. The process of producing many temporal segment candidates is computationally expensive, even

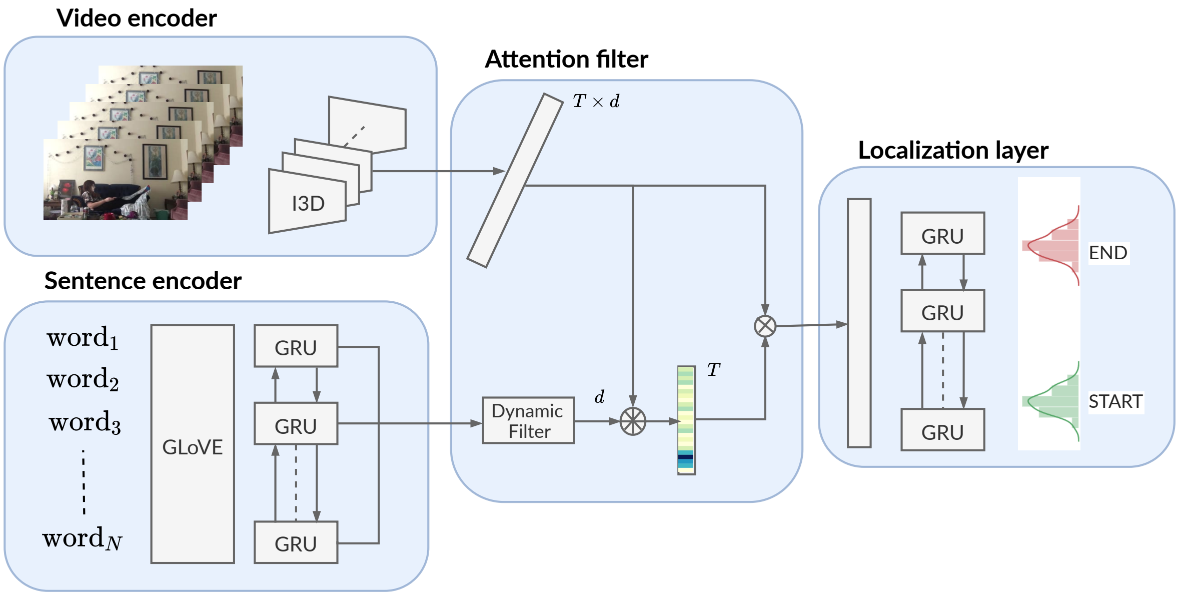

Figure 2: Overview of our method with its four modules: sentence and video encoders to extract features from each modality; a dynamic filter to transfer language information to video, and a localization layer to the starting and ending points.

though its efficiency can be improved if the proposals are processed in parallel. Moreover, proposal-based mechanisms neglect time dependencies across segments, treating them independently thus ultimately failing to effectively capture the temporal information in the input video.

Inspired by recent works in one-shot object detection, we propose a video encoding layer that generates a visual representation summarizing spatio-temporal patterns directly from the raw input frames. Concretely, given an input video V, let $F_V(V)$ be our video encoding function mapping the l input video frames to a sequence of vectors $G = \{g_i \in \mathbb{R}^{d_v}\}$, $i = 1, \dots, n$, with features that capture high-level visual semantics in the video. Note that the length of the input vector in frames l and the number of output features n may differ, which is why we denote them differently.

Because of the encoding of the video, the location of the annotated natural language description needs to be re-scaled to match the new feature-wise setting. We apply the mapping $\tau = (t \cdot n \cdot fps)/l$ to convert from frame/feature index to time. Concretely, t^s and t^e are converted into τ^s and τ^e corresponding to specific integer feature positions such that $\tau^s, \tau^e \in [1, \dots, n]$.

Specifically, in this work we model F_V using I3D [4]. This method inflates the 2D filters of a well-known network e.g. Inception [46, 22] or ResNet [18] for image classification to obtain 3D filters, helping us better exploit the spatio-temporal nature of video. However, note that our video encoder later is generic, so other alternatives such as C3D [48] could be utilized instead.

3.2. Sentence Encoder

The language encoder aims at generating a semantically rich representation of the natural language query that is useful for localizing relevant moments in the video. We model our encoder as a function $F_S(S)$ that maps each word s_j $j = 1, \dots, m$ to a semantic embedding vector $h_j \in \mathbb{R}^{d_s}$, where d_s defines the hidden dimension of the obtained sentence representation.

Although our sentence encoding module is generic, in this work we rely on a bi-directional GRU [9] on top of pre-trained word embeddings. Specifically, we make use of GloVe [34], which are vectors pre-trained in a large collection of text documents. In this setting, our query encoding function F is parametrized by both the GloVe embeddings and the GRU. Finally, to obtain a fixed-length sentence representation we utilize a mean pooling layer over the hidden states obtained from the GRU, obtaining a final summarized query representation $\bar{h}$.

3.3. Guided Attention.

After encoding both the input sentence and video we utilize an attention-based *dynamic filter* [23, 28, 14, 54]. The motivation behind this is to allow the model to generate filters to be applied over the video features that dynamically change given the sentence query, effectively reacting to specific parts of the video embedding and thus providing the model with clues about the location.

Concretely, we first reduce the dimensionality of the sentence embedding d^s and the video embedding d^v to the same space of size d using a fully connected network, and apply a filter function θ as follows.

$$\theta(x) = tanh(W_\theta x + b_\theta) \in \mathbb{R}^d \tag{1}$$

As seen in Equation 1, our filter function $\theta(\cdot)$ is a single-layer fully-connected neural network. The sentence representation $\bar{h}$ is fed into our function and the obtained filter is later used to create a temporal attention over the video features G. Specifically, we apply a softmax over the dot product between each video feature g_i and the output of the filter $\theta(\bar{h})$, as follows.

$$A = softmax\left(\frac{G^\intercal \theta(\bar{h})}{\sqrt{n}}\right) \in \mathbb{R}^n \tag{2}$$

$$\bar{G} = A \odot G \in \mathbb{R}^{n \times d} \tag{3}$$

where $\odot$ denotes the Hadamard product, and the $1/\sqrt{n}$ constant is used to re-scale the product for better training stability [49]. As a result of these operations, each video feature is scaled by the attention filter based on the natural language query.

Given a category of semantically similar natural language queries, for example describing the same type of action, we would like our model to focus on the spatio-temporal features that most likely describe and generalize these semantics across all examples where they are relevant, regardless of the additional context in the videos. We therefore argue that the most relevant features should fall inside the time boundary (τ_s to τ_e) defined by the starting and ending points of the target locations to be predicted. Although features from outside this segment could also contain useful information for the localization task, we hypothesize that by exploiting these features the model should attain less generalization power, as these features are not likely to capture patterns that appear in the majority of different videos containing a given type of action.

In light of this, we encourage our model to attend these relevant features and therefore improve its generalization capabilities. Concretely, we guide our attention mechanism to put focus on these features using a loss function on the output, as follows.

$$L_{att} = -\sum_{i=1}^{n}(1 - \delta_{\tau^s \leq i \leq \tau^e}) \log(1 - a_i) \tag{4}$$

where δ is the Kronecker delta and a_i is the ith column in the attention matrix A.

3.4. Localization Layer

The localization layer is in charge of predicting the starting and ending points of the moment in the video, using the previously constructed sequence of attended video features $\bar{g}_i$ $i = 1, \ldots, n$.

Humans have difficulty agreeing on the starting and ending time of an action inside a video, as evidenced by the low inter-annotation agreement in the relevant datasets for temporal localization [42, 1]. Considering that this is therefore a highly subjective task, we take a probabilistic approach and propose to use *soft-labels* [39, 47] to model the uncertainty associated to the labels.

The localization layer starts by further contextualizing the attended video features $\bar{g}_i$ utilizing a 2-layer bidirectional GRU [9]. Then, we utilize two different fully connected layers to produce scores associated to the probabilities of each position being the start/end of the location. For each case, we take the softmax of these scores and thus obtain vectors $\hat{\boldsymbol{\tau}}^s, \hat{\boldsymbol{\tau}}^e \in \mathbb{R}^n$ containing a categorical probability distribution for the predicted start and end positions, respectively.

To model annotation uncertainty, we take τ^s and τ^e and create two target categorical distribution vectors $\boldsymbol{\tau}^s \sim \mathcal{N}(\tau^s, 1) \in \mathbb{R}^n$ and $\boldsymbol{\tau}^e \sim \mathcal{N}(\tau^e, 1) \in \mathbb{R}^n$ respectively, where $\mathcal{N}(\mu, \sigma)$ denotes a quantized Gaussian distribution centered at μ, with standard deviation σ —one value for each feature. We train our model to minimize the Kullback-Leibler divergence between the predicted and ground truth probability distributions, as follows.

$$L_{KL} = D_{\text{KL}}(\hat{\boldsymbol{\tau}}^s \parallel \boldsymbol{\tau}^s) + D_{\text{KL}}(\hat{\boldsymbol{\tau}}^e \parallel \boldsymbol{\tau}^e) \tag{5}$$

where D_{KL} is the Kullback-Leibler divergence. The final loss for training our method is the sum of the two individual losses defined previously.

$$Loss = L_{KL} + L_{att} \tag{6}$$

During inference, we predict the starting and ending positions using the most likely locations given by the estimated distributions:

$$\hat{\tau}^s = \arg\max(\hat{\boldsymbol{\tau}}^s) \quad \hat{\tau}^e = \arg\max(\hat{\boldsymbol{\tau}}^e) \tag{7}$$

These values correspond to positions in the feature domain of the video, so we convert them back to time positions as explained previously.

4. Experiments

In this section, we first describe the datasets used in our experiments and give some details about our learning procedure. Then, we provide an ablation study on the effect of different components of our approach containing soft-labeling and guided attention and we compare our approach to the state-of-the-art methods. Finally, we provide a qualitative visualization of the predicted localization and attention.

4.1. Datasets

To evaluate our proposed approach we work with two challenging datasets for temporal natural language-driven moment localization, Charades-STA [12] and ActivityNet Caption [3, 26], both of which are widely utilized in previous works for evaluating models on our task.

Charades-STA: built upon the Charades dataset [43] which provides time-based annotations using a pre-defined set of activity classes, and general video descriptions. In [12], the sentences describing the video are semi-automatically decomposed into smaller chunks and aligned with the activity classes, which are later verified by human annotators. As a result of this process, the original class-based activity annotations are effectively associated to their natural language descriptions, totalling 13,898 pairs. We use the predefined train and test splits, containing 12,408 and 3,720 moment-query pairs respectively. Videos are 31 seconds long on average, with 2.4 moments on average, each being 8.2 seconds long on average.

ActivityNet Caption (ANet-Cap): a large dataset built on top of ActivityNet [3], which contains data derived from YouTube and annotated for the tasks of activity recognition, segmentation and prediction. ANet-Cap further improves the annotations in ANet by incorporating descriptions for each temporal segment in the videos, totalling up to 100K temporal descriptions annotations over 20K videos. These have an average length of 2.5 minutes and are associated to over 200 activity classes, making the content much more diverse compared to Charades-STA. The temporally annotated moments are 36 seconds long on average, with videos containing 3.5 moments on average. Besides moments being longer than in Charades-STA, we find that their associated natural language descriptions are also longer, besides using a more varied and richer vocabulary. We utilize the predefined train and validation splits in our experiments. Unlike Charades, Activity-Net contains a moment covering the entire video.

Although other similar datasets, such as DiDeMo [19] and TACoS [37] also exist, we find them inadequate for evaluating our method. In the case of DiDeMo, we note this dataset has been constructed for purposes that are substantially different from ours, lacking start/end temporal annotations. In the case of TACoS, although it shares some similarities with ANet-Cap, we find that several training/evaluation splits exist, with different previous work adopting different alternatives. Therefore, comparisons against these approaches would be less meaningful and not very informative. At the same time, as its contents are derived entirely from a single video topic –cooking scenes– this dataset appears less challenging when compared to our considered alternatives.

Method	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
NLL	60.91	43.66	27.07
KL	66.69	47.20	29.35
NLL + AL	66.64	47.53	29.89
KL + AL	**67.53**	**52.02**	**33.74**

Table 1: Ablation study on the impact of the guided attention and soft-labeling on Charades-STA.

4.2. Implementation Details

Pre-processing for the natural language input in the case of Charades-STA is minimal, as we simply tokenize and keep all the words in the training data. In the case of ANEt-Cap, we pre-process the text with spacy[2] and replace all named entities as well as proper nouns with special markers. Finally, we truncate all sentences to a maximum length of 30 words and replace all tokens with frequency lower than 5 in the corpus with a special *UNK* marker. The language encoder uses a hidden state of size 256, without fine-tuning the pre-trained GloVe embeddings.

When it comes to the video encoder, we first pre-process the videos by extracting features of size 1024 using I3D with average pooling, taking as input the raw frames of dimension 256×256, at 25 fps. For Charades-STA, we use the pre-trained model released by [4] trained on Charades. For Anet-Cap we use the model pre-trained on the kinetics400 dataset [24] released by the same authors, which we also fine-tune on ANet-Cap afterwards.

All of our models are trained in an end-to-end fashion using Adam [25] with a learning rate of 10^{-4} and weight decay 10^{-3}. To prevent over-fitting, we add dropout 0.5 between the two layers in the localization module, which has a hidden size of 256. In addition to this, we also apply a simple data augmentation technique that consists on creating new examples by randomly cropping segments out from the initial part of the videos. We do this whenever the random cropping does not overlap with the locations of the annotations.

4.3. Evaluation Metric

We evaluate our model by computing the temporal Intersection over Union (tIoU) at different thresholds, which we denote using the α parameter. In this setting, for a given value of α, whenever a given predicted time window has an intersection with the gold-standard that is above the α threshold, we consider the output of the model as correct. Following previous work, we also report the mean tIoU (mIoU) on the ANet-Cap dataset, helping make our comparisons more comprehensive.

[2] https://spacy.io

4.4. Ablation Study

To show the effectiveness of some introduced components, we perform several ablation studies on the Charades-STA dataset. Concretely, we focus on the soft-labeling technique and the usage of the attention loss L_{att}. For the latter we simply experiment omitting the term for the calculation of the gradients. For the former, we replace the L_{KL} loss with a likelihood-based loss similar to [16], as follows:

$$L_{NLL} = -\log(\hat{\boldsymbol{\tau}}^s[\tau^s]) - \log(\hat{\boldsymbol{\tau}}^e[\tau^e]) \tag{8}$$

where $\hat{\boldsymbol{\tau}}^s$ and $\hat{\boldsymbol{\tau}}^e$ are the predicted probability distributions and τ^s and τ^e are the respective indices from the ground-truth annotations.

We first compare our *soft-labeling* approach with the previously mentioned likelihood-based loss (NLL). As shown in Table 1, modeling the subjectivity of the labeling process using soft-labels and our distribution-matching loss (KL) leads to a significant improvement in the performance of our method, both in terms of retrieving and localizing the full extent of the queries in the given videos.

We also evaluate the contribution of the attention loss L_{att} to our pipeline. According to the results in Table 1, adding the attention loss (AL) leads to a consistent improvement in the performance of our method, both when modeling soft-labels and when not. This confirms our hypothesis that the most generalizable features are likely to be located within the boundaries of the query segment in the video. Finally, the synergy of our two proposed techniques can be seen in the last row of Table 1.

4.5. Comparison to the State-of-the-Art

We compare the performance of our proposed approach on both datasets against several prior work baselines. **Proposal-based methods**: We compare our approach to a broad selection of models based on proposal generation, including MCN [19], TGN [7], MAN [54], as well as some recent work such as SAP [8], MLVI [52] and ACRN [31]. **Reinforcement-learning-based methods**: We compare our results to TripNet [17] and SMRL [50], both of which utilize RL to learn how to jump through the video until the correct localization is found.
Proposal-free methods: We consider two recent works, ABLR [53] and ExCL [16], both aiming for proposal-free moment localization. Similar to ours, these techniques utilize the complete video representation to predict the start and end of a relevant segment. However, our approach is different since it models the uncertainty of the labeling process. Note also that while ABLR utilizes a co-attention layer, ExCL does not rely on attention layers at all.

Comparing the performance of our method in the **Charades-STA** benchmark, our proposed approach outperforms all the baselines by a large margin, as can be seen in Table 2. Its mean temporal intersection over union is 48.22 reflecting the capability of our method to correctly identify the correct temporal extent of the natural language query. As can also be seen in the performance at $\alpha = 0.7$ and $\alpha = 0.9$ where our method obtains 33.74 and 9.68 accuracy for those thresholds.

Method	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
Random	-	8.51	3.03
CTRL [12]	-	21.42	7.15
ABLR [8]	-	24.36	9.01
SMRL[50]	-	24.36	11.17
SAP [8]	-	27.42	13.36
MLVI [52]	54.70	35.60	15.80
TripNet [17]	51.33	36.61	14.50
ExCL [16]	65.10	44.10	23.30
MAN [54]	-	46.53	22.72
Ours	**67.53**	**52.02**	**33.74**

Table 2: Accuracy on Charades-STA for different tIoU α levels. Results for ABLR are as reported by [8].

Method	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
MCN [19]	42.80	21.37	9.58	-
CTRL [12]	49.09	28.70	14.00	-
ACRN [53]	50.37	31.29	16.17	-
MLVI [52]	-	45.30	27.70	13.60
TGN [7]	70.06	45.51	28.47	-
TripNet [17]	-	48.42	32.19	13.93
ABLR [53]	73.30	**55.67**	**36.79**	-
Ours	**75.25**	51.28	33.04	**19.26**

Table 3: Accuracy on ANet-Cap for different tIoU α levels.

ANet-Cap is a challenging benchmark not only because the length of the video is much longer but also because it presents a bigger variability of the segment duration for a query than Charades-STA. Since we are not processing videos using proposals this types of videos could present difficulties to our localization layer. However, as shown in Table 3, our method yields good performance at different levels of tIoU. In particular, it outperforms all previous methods at α 0.1 and 0.7, showing the effectiveness of our method to recall the correct temporal extent of sentence query. Although our method cannot outperform the performance of ABLR at α 0.3 and 0.5, it yields better mIoU than previous methods in this benchmark, as can be seen in Table 4. It is important to note that in this case we do not compare with ExCL [16] since their reported results have more than 3,300 missing videos.

As suggested by the empirical evidence, our method consistently outperforms others on estimating the correct extension of the queries. This indicates that our proposed mechanism for incorporating the uncertainty of the labeling process is effective yet simple, playing a key role on helping

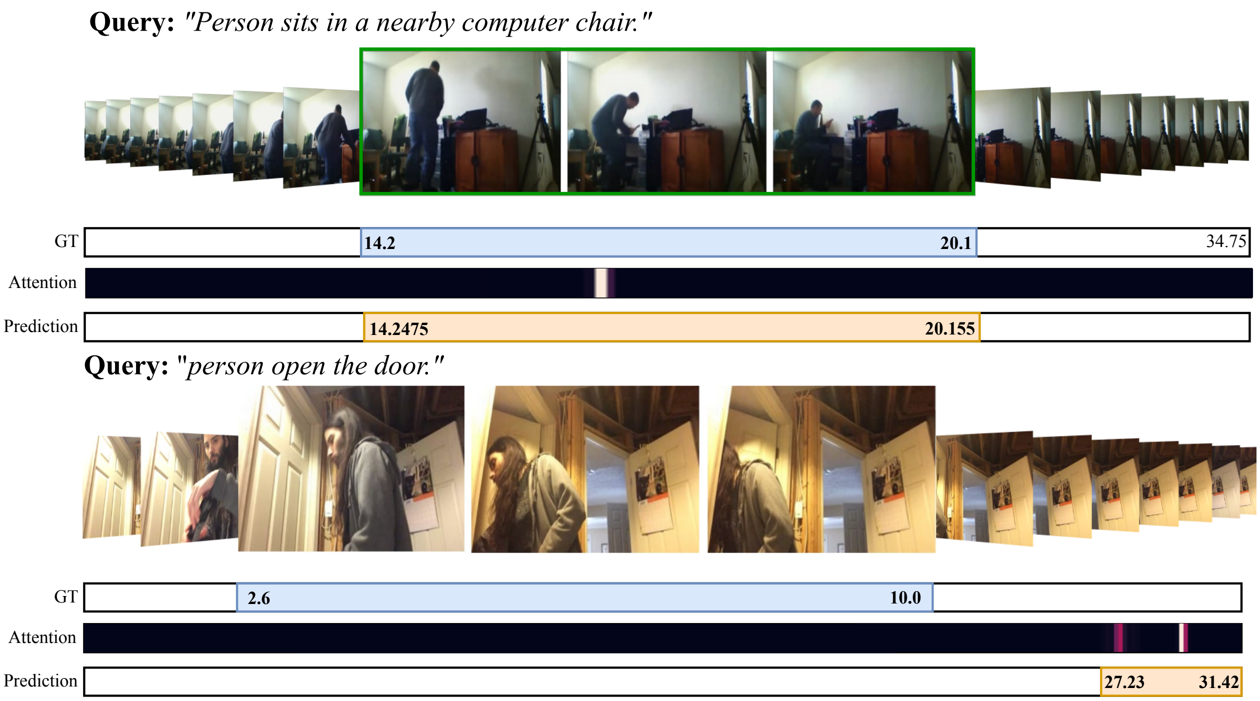

Figure 3: Examples of success and failure cases of our method for Charades-STA.

Method	MCN	CTRL	ACRN	ABLR	Ours
Mean tIoU	15.83	20.54	24.16	36.99	37.78

Table 4: Mean tIoU in the ANet-Cap benchmark.

the network to find the correct starting and ending points. In addition to this, the evidence also suggest that our novel attention mechanism, which guides the localization layer to focus on the features that are within the corresponding segments in the video also aids the network. By allowing the model to attend the features that better represent similar action across different videos, we obtain better generalization.

4.6. Qualitative Results

Two different samples, one showing a success and one a failure case of our method on Charades-STA dataset can be seen in Figure 3. Each sample presents the ground truth localization, the attention weights and predicted localization of a given query. For the attention, brighter colors mean more weight. In the successful case, given the query *“Person sits in a nearby computer chair.”* our method could localize the moment at a tIoU of 98.28%, with a maximum attention at 16.27 seconds peaking at 0.83. It is interesting to see that only one or two video features seem to be necessary for retrieving the starting and ending correctly.

On the second example in Figure 3 we show how our method fails to localize the query *“person open the door”*. It is possible to see that the appearance of the retrieved moment, when the person actually leaves the room, is very similar to the ground truth, Figure 4. We attribute this result to the features for opening the door and leaving the room being too close, especially on this example. Probably high quality spatio-temporal features or deeper reasoning about the context would help the network to disambiguate this type of scenarios.

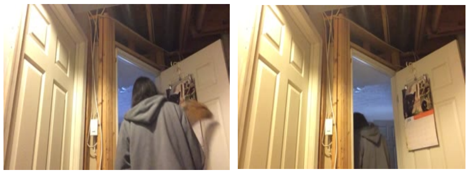

Figure 4: Similar appearance frames for failure case on Charades-STA

5. Conclusion

In this paper we have presented a novel end-to-end architecture that is designed to address the problem of temporal localization of natural-language queries in videos. Our approach uses a guided attention mechanism that focus on more generalizable features to guide the localization estimation. Moreover, we also consider the key problem of subjectivity in the annotation process by modeling the label uncertainty in a simple but efficient way, also obtaining substantial performance gains. As a result, our approach archives state-of-the-art performance on both Charades-STA and ANet-Cap datasets.

References

[1] H. Alwassel, F. Caba Heilbron, V. Escorcia, and B. Ghanem. Diagnosing error in temporal action detectors. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 5

[2] H. Alwassel, F. Caba Heilbron, V. Escorcia, and B. Ghanem. Diagnosing Error in Temporal Action Detectors. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision ECCV 2018*, volume 11207, pages 264–280. Springer International Publishing, Cham, 2018. 2

[3] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 6

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 4, 6

[5] Y. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster R-CNN architecture for temporal action localization. *CVPR*, 2018. 1

[6] D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading Wikipedia to Answer Open-Domain Questions. pages 1870–1879, July 2017. 2

[7] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua. Temporally grounding natural sentence in video. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 162–171, Brussels, Belgium, 2018. Association for Computational Linguistics. 3, 7

[8] S. Chen and Y.-G. Jiang. Semantic proposal for activity localizaiton in videos via sentence query. *AAAI*, 2019. 3, 7

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4, 5

[10] V. Escorcia, F. Caba Heilbron, J. C. Niebles, and B. Ghanem. DAPs: Deep Action Proposals for Action Understanding. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision ECCV 2016*, Lecture Notes in Computer Science, pages 768–784. Springer International Publishing, 2016. 2

[11] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. DAPs: Deep Action Proposals for Action Understanding. *ECCV*, 2016. 1

[12] J. Gao, C. Sun, Z. Yang, and R. Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017. 1, 2, 3, 6, 7

[13] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. TURN TAP: temporal unit regression network for temporal action proposals. *ICCV*, 2017. 1

[14] K. Gavrilyuk, A. Ghodrati, Z. Li, and C. G. M. Snoek. Actor and action video segmentation from a sentence. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[15] R. Ge, J. Gao, K. Chen, and R. Nevatia. Mac: Mining activity concepts for language-based temporal localization. In *WACV*, 2019. 2, 3

[16] S. Ghosh, A. Agarwal, Z. Parekh, and A. Hauptmann. Excl: Extractive clip localization using natural language descriptions. *arXiv preprint arXiv:1904.02755*, 2019. 2, 3, 7

[17] M. Hahn, A. Kadav, J. M. Rehg, and H. P. Graf. Tripping through time: Efficient localization of activities in videos. *arXiv preprint arXiv:1904.09936*, 2019. 3, 7

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 4

[19] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 1, 2, 3, 6, 7

[20] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018. 3

[21] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The THUMOS challenge on action recognition for videos in the wild. *Computer Vision and Image Understanding*, 155:1–23, Feb. 2017. 2

[22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 4

[23] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pages 667–675, 2016. 4

[24] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, 2017. 6

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. 6

[26] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 2, 6

[27] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 3

[28] Z. Li, R. Tao, E. Gavves, C. G. Snoek, and A. W. Smeulders. Tracking by natural language specification. In *CVPR*, pages 6495–6503, 2017. 4

[29] T. Lin, X. Zhao, and Z. Shou. Single Shot Temporal Action Detection. *ACMMM*, 2017. 1

[30] T. Lin, X. Zhao, and Z. Shou. Single Shot Temporal Action Detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, pages 988–996, New York, NY, USA, 2017. ACM. event-place: Mountain View, California, USA. 2

[31] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua. Attentive moment retrieval in videos. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 15–24. ACM, 2018. 3, 7

[32] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. 2019. 1

[33] S. Ma, L. Sigal, and S. Sclaroff. Learning Activity Progression in LSTMs for Activity Detection and Early Detection. pages 1942–1950, 2016. 2

[34] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 4

[35] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3

[36] A. Richard, H. Kuehne, A. Iqbal, and J. Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, 2018. 1

[37] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In X. Jiang, J. Hornegger, and R. Koch, editors, *Pattern Recognition*, 2014. 6

[38] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A Dataset for Movie Description. pages 3202–3212, 2015. 2

[39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 5

[40] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[41] G. A. Sigurdsson, O. Russakovsky, and A. Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, 2017. 2

[42] G. A. Sigurdsson, O. Russakovsky, and A. Gupta. What actions are needed for understanding human actions in videos? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2137–2146, 2017. 5

[43] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, 2016. 6

[44] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 2

[45] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection. pages 1961–1970, 2016. 2

[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 4

[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. pages 2818–2826, 2016. 5

[48] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2, 4

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 5

[50] W. Wang, Y. Huang, and L. Wang. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 334–343, 2019. 3, 7

[51] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. *ICCV*, 2017. 1

[52] H. Xu, K. He, L. Sigal, S. Sclaroff, and K. Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*, 2019. 3, 7

[53] Y. Yuan, T. Mei, and W. Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. *AAAI*, 2019. 3, 7

[54] 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. *CVPR*, 2019. 3, 4, 7

[55] Y. Zhao, Y. Xiong, L. Wang, Z. W. . . . . V. (ICCV), and U. 2017. Temporal action detection with structured segment networks. *ICCV*, 2017. 1