# Read, Watch, and Move: Reinforcement Learning for Temporally Grounding Natural Language Descriptions in Videos

**Dongliang He,[†] Xiang Zhao,[†] Jizhou Huang,[‡,†] Fu Li,[†] Xiao Liu,[†] Shilei Wen[†]**
[†]Baidu Inc., Beijing, China
[‡]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
{hedongliang01, zhaoxiang05, huangjizhou01, lifu, liuxiao12, wenshilei}@baidu.com

## Abstract

The task of video grounding, which temporally localizes a natural language description in a video, plays an important role in understanding videos. Existing studies have adopted strategies of sliding window over the entire video or exhaustively ranking all possible clip-sentence pairs in a pre-segmented video, which inevitably suffer from exhaustively enumerated candidates. To alleviate this problem, we formulate this task as a problem of sequential decision making by learning an agent which regulates the temporal grounding boundaries progressively based on its policy. Specifically, we propose a reinforcement learning based framework improved by multi-task learning and it shows steady performance gains by considering additional supervised boundary information during training. Our proposed framework achieves state-of-the-art performance on ActivityNet'18 DenseCaption dataset (Krishna et al. 2017) and Charades-STA dataset (Sigurdsson et al. 2016; Gao et al. 2017) while observing only 10 or less clips per video.

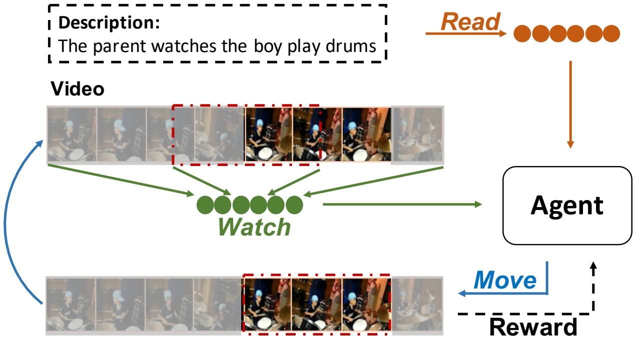

Figure 1: Illustration of the proposed formulation of read, watch, and move with reinforcement learning. At each time step, the agent makes an observation, takes an action, and receives a reward evaluating that action. The policy is updated according to this reward.

## 1 Introduction

The task of grounding natural language in image/video, which aims to localize a short phrase (Rohrbach et al. 2016; Endo et al. 2017; Tellex and Roy 2009; Regneri et al. 2013) or a referring expression (Bojanowski et al. 2015; Gao, Sun, and Nevatia 2016) in a given image/video, is essential for understanding the way human perceive images and scenes. In the past few years, studies on detecting a particular object in image (Pirinen and Sminchisescu 2018) or activity in video (Yeung et al. 2016; Singh et al. 2016; Zhao et al. 2017; Shou et al. 2017) have been extensively exploited. However, they are restricted to a predefined close set of object/activity space. In this paper, we take a further step by focusing on the problem of temporally grounding an open-ended natural language description in a long, untrimmed video.

Specifically, given an untrimmed video and a natural language description, our task is to determine the start and end time stamps of a clip in this video that corresponds to the given description. Exploration on natural language enables us to deal with not only an open set of objects/activities, but also the correlations and interactions between the involved entities. This is a challenging task as it requires both vision and language understanding, but it has important applications in video understanding tasks such as video retrieval and text-oriented video highlight detection. Despite the fact that both action localization and video grounding tasks share the same goal of extracting a well-matched video clip w.r.t. a given query, the methods for action localization are not readily applicable to video grounding. The main reason is that it requires to map natural language descriptions to a close set of labels before applying these methods, which inevitably causes information loss. For example, if the sentence “The parent watches the boy play drums” is mapped to an action label “play drums”, action detectors can hardly tell the start point of the presence of the parent.

Although there is a growing interest in grounding natural language in videos, e.g., (Gao et al. 2017; Hendricks et al. 2017; Liu et al. 2018), these methods have a limitation that they can be applied only when all candidates are exhaustively enumerated, as they need to slide over the entire video or rank all possible clip-sentence pairs to be successful in generating the final grounding result. Moreover, ranking strategy (Hendricks et al. 2017) is specially designed for pre-segmented videos and is not applicable to general cases. To effectively detect the temporal boundaries of a given description from a video, especially from a long, untrimmed video, it is important to make judgments only by several glimpses. To the best of our knowledge, the problem of grounding natural language in videos without sliding over the entire video has not been well addressed.

In this paper, we propose an end-to-end reinforcement learning (RL) based framework for grounding natural language descriptions in videos. As shown in Figure 1, the agent iteratively reads the description and watches the entire video as well as the temporary grounding clip and its boundaries, and then it determines where to move the temporal grounding boundaries based on the policy. At each time step, the agent makes an observation that details the currently selected video clip, and takes an action according to its policy. The environment is updated accordingly and offers the agent a reward that represents how well the chosen action performs. Our key intuition is that given a description and the current grounding start and end points, human can make hypotheses on how these points should be moved after observing a clip, and then iteratively adjust the grounding boundaries step by step towards finding the best matching clip. Therefore, we naturally formulate this task as learning an agent for sequential decision making to regulate the temporal boundaries progressively based on the learned policy. In addition, we introduce two distinctive characteristics into the RL framework. First, we improve the framework with multi-task learning and show steady performance gains by considering additional supervised boundary information during training. Second, we introduce a location feature, which enables the agent to explicitly access where the current grounding boundaries are.

Specifically, we design an observation network that takes visual and textual information as well as current temporal boundaries into consideration. Given the observation, the decision making policy is modeled by a policy network that outputs the probability distribution over all possible actions, and then the agent adjusts the grounding boundaries by taking an action according to the policy. The policy network is implemented using a recurrent neural network (RNN). In addition, in order to achieve better feature learning, we propose to leverage the ground truth for supervised temporal IoU (tIoU) regression as well as boundary regression at each step. Our RL model can be trained end-to-end for natural language description grounding within several steps of glimpses (observations) and temporal boundaries adjustments (actions). We evaluate our model on two well-known datasets, i.e., ActivityNet DenseCaption (Krishna et al. 2017) and Charades-STA (Sigurdsson et al. 2016; Gao et al. 2017). Experimental results show that our method achieves state-of-the-art performance on these datasets and validate the superiority of our method.

Here we summarize our main contributions:

- We propose a reinforcement learning based framework for the task of grounding natural language descriptions in videos.
- Supervised learning is combined with reinforcement learning in a multi-task learning framework, which helps the agent obtain more accurate information about the environment and better explore the state space to encounter more reward, thus it can act more effectively towards task accomplishment.
- Experiments conducted on two well-known datasets demonstrate that the proposed method significantly outperforms state-of-the-art method by a substantial margin, which verifies the effectiveness of our method.

## 2 Related Work

We first introduce the related work of grounding natural language in visual content. Then, we introduce the related work on applying reinforcement learning for vision tasks.

The problem of grounding natural language descriptions in videos has attracted an increasing interest more recently. Previous studies typically treat it as a regression or ranking problem by adopting strategies of sliding window over the entire video (Gao et al. 2017; Liu et al. 2018) or exhaustively ranking all possible clip-sentence pairs in a pre-segmented video (Hendricks et al. 2017). However, it is difficult to attain a reasonable compromise between accuracy and computational efficiency due to they inevitably suffer from exhaustively enumerated candidates and the false positive problem. To alleviate this problem, we formulate the task as a sequential decision making problem, rather than treat it as a regression or ranking problem. To this end, we propose an effective method to address this problem with a reinforcement learning based framework.

Our work is also related to the work on understanding visual content. Here we briefly review the closely related work including image grounding, image/video captioning, video retrieval, and temporal action localization in videos. The task of image grounding, which aims to localize a short phrase or an expression in an image, is a well-studied field, e.g., (Plummer et al. 2015; Rohrbach et al. 2016; Xiao, Sigal, and Jae Lee 2017; Endo et al. 2017). However, they have a different focus from ours. The task of image/video captioning, which aims to generate a description in text from visual data, has been well studied in computer vision, e.g., (Xu et al. 2015a; Pan et al. 2016; Gan et al. 2017; Liang et al. 2017). The task of visual grounding can be regarded as the conjugate task of visual description generation. Sentence based video retrieval methods (Xu et al. 2015b; Lin et al. 2014; Bojanowski et al. 2015) use skip-thought (Kiros et al. 2015) or bidirectional LSTM to encode text queries, and incorporate video-language embedding to find the video that is most relevant to the query. They mainly focus on retrieving an entire video for a given query, rather than finding a temporal segment that matches a given description from a video. The task of temporal action localization is defined as retrieving video segments relevant to a given action from a predefined close set of action labels (Singh et al. 2016; Lin, Zhao, and Shou 2017; Zhao et al. 2017; Yeung et al. 2016). There is also some work on retrieving temporal segments relevant to a sentence within a video in constrained settings. For example, Tellex and Roy (2009) studied the problem of retrieving video clips from home surveillance videos using text queries from a fixed set of spatial prepositions. Bojanowski et al. (2015) proposed to align video seg-

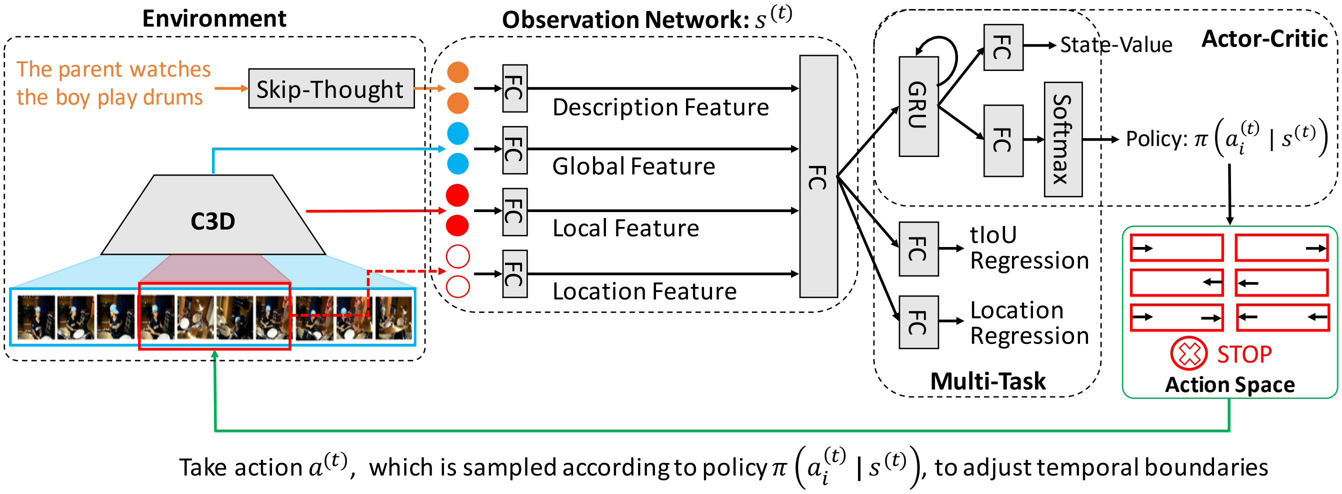
Environment
The parent watches the boy play drums
Skip-Thought
C3D
Observation Network: $s^{(t)}$
FC
Description Feature
Global Feature
Local Feature
Location Feature
GRU
State-Value
Actor-Critic
Softmax
Policy: $\pi\left(a_i^{(t)} \mid s^{(t)}\right)$
tIoU Regression
Location Regression
Multi-Task
STOP
Action Space
Take action $a^{(t)}$, which is sampled according to policy $\pi\left(a_i^{(t)} \mid s^{(t)}\right)$, to adjust temporal boundaries

Figure 2: Architecture of the RL-based framework. The action space consists of 7 different ways to adjust the temporal boundaries. State vector $s^{(t)}$, which fuses description embedding feature, global video feature, local video feature, and current location embedding feature, is fed into the actor-critic (Sutton and Barto 2011) to learn both a policy and a state-value. Temporal IoU between current video clip and the ground truth is regressed with supervised learning, so are the temporal localization boundaries.

ments with a set of sentences which is temporally ordered. Our work is different in that we temporally ground open-ended, arbitrary natural language descriptions in videos.

The use of reinforcement learning (Williams 1992) in our task is inspired by the recent remarkable success of applying it in various vision tasks. The goal of reinforcement learning is to learn an agent to evaluate certain actions under particular states or learn an action policy with respect to current states. It is effective to optimize the sequential decision making problems. Recently, it has been successfully applied to learn task-specific policies in various computer vision problems. For example, spatial attention policies are learned for image recognition in (Mnih et al. 2014) and (Liu et al. 2017). RL framework is also leveraged for object detection in (Pirinen and Sminchisescu 2018). Li et al. (2018) proposed an aesthetic aware reinforcement learning framework to learn the policy for image cropping and achieved promising results. As for vision tasks in video, (Yeung et al. 2016) proposed to use RL for action localization in videos and achieved a good trade-off between localization accuracy and computation cost. Recently, video object segmentation with deep reinforcement cutting-agent learning is explored in (Han et al. 2018), and hierarchical reinforcement learning (Sutton and Barto 2011) is incorporated for video description in (Wang et al. 2018). Our video grounding task can be formulated as a sequential decision making problem, thus it fits naturally into the reinforcement learning paradigm. While conceptually similar, our model is novel in that it successfully combines supervised learning with reinforcement learning in a multi-task learning framework, which helps the agent obtain more accurate information about the environment and better explore the state space to encounter more reward, thus it can act more effectively towards task accomplishment.

## 3 Methodology

Given the insight that human usually localize a description in a video by following the iterative progress of observing a clip, making a hypothesis on how the boundaries should be shifted, and verifying it, we formulate the task of temporally grounding descriptions in videos as a sequential decision-making problem. Therefore, it naturally fits into the RL framework. Figure 2 shows the architecture of the proposed model. The temporal grounding location is initially set to $[l_s^{(0)}, l_e^{(0)}]$. At each time step, the observation network outputs the current state of the environment for the actor-critic module to generate an action policy. The policy defines the probabilistic distribution of all the actions in the action space and can be easily implemented by softmax operation. The agent samples an action according to the policy and adjusts the temporal boundaries iteratively until encountering a STOP action or reaching the maximum number of steps (denoted by $T_{max}$).

### 3.1 Environment and Action Space

In this task, the action space $\mathcal{A}$ consists of 7 predefined actions, namely, *moving start/end point ahead by* $\delta$ , *moving start/end point backward by* $\delta$, *shifting both start and end point backward/forward by* $\delta$, and a *STOP* action. In this paper, $\delta$ is set to $N/10$, where $N$ is the video length. In fact, the two shifting actions can be accomplished by two successive moving actions, and we define them mainly for the purpose of enabling a faster temporal boundary adjustment.

Once an action is executed, the environment changes accordingly. Environment here is referred to as the combination of the video, the description, and the current temporal grounding boundaries. We use skip-thought (Kiros et al. 2015) to generate the sentence embedding $E$ and feed the entire video into C3D (Tran et al. 2015) to obtain the global feature vector $V_G$. At each step, these features are retained. To represent the status of the current temporal grounding boundaries for the $t^{th}$ step of decision, we propose to explicitly involve the normalized start point and end point $L^{(t-1)} = [l_s^{(t-1)}, l_e^{(t-1)}], t = 1, 2, ..., T_{max}$ as supplementary to the local C3D feature $V_L^{(t-1)}$ of the current video clip. In the observation network, sentence embedding, global video feature, local video feature, and normalized temporal boundaries are encoded using fully-connected (FC) layers and their outputs are concatenated and fused by a FC layer to generate the state vector $s^{(t)}$, which is calculated as:

$$s^{(t)} = \Phi(E, V_G, V_L^{(t-1)}, L^{(t-1)}). \tag{1}$$

## 3.2 Advantage Actor-Critic RL Algorithm

It has been proven that actor-critic algorithm (Sutton and Barto 2011) is able to reduce the variance of the policy gradient. As shown in Figure 2, the sequential decision making process is modeled by an actor-critic module, which employs GRU (Cho et al. 2014) and FC layers to output the policy $\pi(a_i^{(t)}|s^{(t)}, \theta_\pi)$ and a state-value $v(s^{(t)}|\theta_v)$. The policy determines the probabilistic distribution of actions and the state-value serves as estimation of the value of the current state (i.e., critic). $\theta_\pi$ denotes parameters of the observation network and the policy branch, $\theta_v$ denotes parameters of the critic branch. The input of the GRU at each time step is the observed state vector $s^{(t)}$. The hidden state of the GRU unit is used for policy generation and state-value estimation.

**Reward** Our goal is to learn a policy that can correctly grounding descriptions in videos, so the reward for the reinforcement learning algorithm should encourage the agent to find a better matching clip step by step. Intuitively, allowing the agent to search within more steps could yield a more accurate result but it will introduce more computational cost. To alleviate this issue, we reward the agent for taking better actions: the agent will get a positive reward 1 if the temporal IoU (tIoU) between $L^{(t)} = [l_s^{(t)}, l_e^{(t)}]$ and the ground-truth grounding location $G = [g_s, g_e]$ gets higher after the $t^{th}$ step of action. On the contrary, we punish the agent for taking worse actions: the agent will get no reward if the tIoU gets lower. To balance the number of actions and the accuracy, we further punish the number of steps by a negative reward of $-\phi * t$, where $\phi$ is the positive penalty factor ranging from 0 to 1. Moreover, there are chances that the start point gets higher than the end point after taking several actions, which is an undesirable state. To avoid such situations, the reward function should punish the agent if the tIoU is a negative value. To sum up, the reward of the $t^{th}$ step is formulated as:

$$r_t = \begin{cases} 1 - \phi * t, & tIoU^{(t)} > tIoU^{(t-1)} \geq 0 \\ -\phi * t, & 0 \leq tIoU^{(t)} \leq tIoU^{(t-1)} \\ -1 - \phi * t, & otherwise \end{cases}, \tag{2}$$

where $tIoU$ is calculated as:

$$tIoU^{(t)} = \frac{min(g_e, l_e^{(t)}) - max(g_s, l_s^{(t)})}{max(g_e, l_e^{(t)}) - min(g_s, l_s^{(t)})}. \tag{3}$$

The task is modeled as a sequential decision making problem, whose goal is to finally hit the ground truth, so the reward of following up steps should be traced back to the current step. Therefore, our final accumulated reward function is designed as follows:

$$R_t = \begin{cases} r_t + \gamma * v(s^{(t)}|\theta_v), & t = T_{max} \\ r_t + \gamma * R_{t+1}, & t = 1, 2, ..., T_{max} - 1 \end{cases}, \tag{4}$$

where $\gamma$ is a constant discount factor.

**Objective** Instead of directly maximizing the expected reward, we introduce the advantage function (Sutton and Barto 2011), and our objective becomes maximizing the expected advantage $J_a(\theta)$:

$$J_a(\theta_\pi) = \sum_{a \in \mathcal{A}} \left\{ \pi(a|s, \theta_\pi)(R - v(s|\theta_v)) \right\}. \tag{5}$$

The gradient of $J_a$ is:

$$\nabla_{\theta_\pi} J_a(\theta_\pi) = \sum_{a \in \mathcal{A}} \pi(a|s, \theta_\pi) \left( \nabla_{\theta_\pi} \log \pi(a|s, \theta_\pi) \right) (R - v(s|\theta_v)). \tag{6}$$

It is difficult to directly optimize this problem due to the high dimension of the action sequence space. In RL, Mente Carlo sampling is adopted and the policy gradient is approximated as:

$$\nabla_{\theta_\pi} J_a(\theta_\pi) \approx \sum_t \nabla_{\theta_\pi} \log \pi(a_i^{(t)}|s^{(t)}, \theta_\pi)(R_t - v(s^{(t)}|\theta_v)). \tag{7}$$

Therefore, the maximizing problem can be optimized by minimizing the following loss function $\mathcal{L}'_A$ using stochastic gradient descent (SGD):

$$\mathcal{L}'_A(\theta_\pi) = -\sum_t \left( \log \pi(a_i^{(t)}|s^{(t)}, \theta_\pi) \right) (R_t - v(s^{(t)}|\theta_v)). \tag{8}$$

We also follow the practice of introducing entropy of the policy output $H(\pi(a_i^{(t)}|s^{(t)}, \theta_\pi))$ into the objective, since maximizing the entropy can increase the diversity of actions (Li et al. 2018). Therefore, the overall loss of the actor branch becomes:

$$\mathcal{L}_A(\theta_\pi) = \mathcal{L}'_A(\theta_\pi) - \sum_t \lambda_0 * H(\pi(a_i^{(t)}|s^{(t)}, \theta_\pi)), \tag{9}$$

where $\lambda_0$ is a constant scaling factor and it is set to 0.1 in our experiments.

In the context of actor-critic algorithm, the objective of the estimated state-value $v(s^{(t)}|\theta_v)$ by critic branch is to minimize the mean squared error (MSE). Therefore, we introduce the MSE loss for the critic branch defined as follows:

$$\mathcal{L}_C(\theta_v) = \sum_t \left( v(s^{(t)}|\theta_v) - R_t \right)^2. \tag{10}$$

Finally, the overall loss of the RL algorithm is a combination of the two losses:

$$Loss_1 = \mathcal{L}_A(\theta_\pi) + \lambda_1 * \mathcal{L}_C(\theta_v). \tag{11}$$

### 3.3 Combination of Supervised Learning and RL

To learn more representative state vectors, we propose to combine reinforcement learning and supervised learning into a multi-task learning framework. The overall loss for supervised learning is a combination of the tIoU regression loss and the location regression loss:

$$Loss_2 = Loss_{tIoU} + \lambda_2 * Loss_{Loc}. \tag{12}$$

Here we define the tIoU regression loss as the L1 distance between the predicted tIoU and the ground-truth tIoU, i.e.,

$$Loss_{tIoU} = \sum_t |tIoU^{(t-1)} - P^{(t)}_{tIoU}|, \tag{13}$$

where $P^{(t)}_{IoU}$ is the predicted tIoU at the $t^{th}$ step. Similarly, the L1 location regression loss can be calculated as:

$$Loss_{Loc} = \sum_t y^{(t)} \left( |g_s - P^{(t)}_s| + |g_e - P^{(t)}_e| \right) /2, \tag{14}$$

where $P^{(t)}_s$ and $P^{(t)}_e$ are the predicted start and end, respectively. $y^{(t)}$ is an indicator which equals 1 when $tIoU^{(t-1)} > 0.4$ and 0 otherwise.

The proposed model can be trained end-to-end by minimizing the overall multi-task loss $Loss_1 + \lambda_3 * Loss_2$. In our experiments, $\lambda_1$, $\lambda_2$, and $\lambda_3$ are empirically set to 1. The supervised loss enables the algorithm to learn to predict how well the current state matches the ground truth and where the temporal boundaries should be regressed while the RL loss enables it to learn a policy for pushing the temporal boundaries to approach the ground truth step by step. In the training phase, parameters are updated with both supervised gradient and policy gradient. Therefore, the generalization of the trained network can be improved by sharing the representations.

## 4 Experiments

### 4.1 Datasets and Evaluation Metric

The models are trained and evaluated on two well-known datasets, i.e., ActivityNet DenseCaption (Krishna et al. 2017) and Charades-STA (Sigurdsson et al. 2016; Gao et al. 2017). The DenseCaption dataset contains 37,421 and 17,505 natural language descriptions with their corresponding ground-truth temporal boundaries for training and test, respectively. The Charades-STA dataset is originally collected for video classification and video captioning. Gao et al. (2017) processed the dataset to make it suitable for video grounding task. In this dataset, there are 13,898 description-clip pairs in the training set, and 4,233 description-clip pairs in the test set.

Following previous work (Yeung et al. 2016; Gao et al. 2017), we also evaluate the models according to the grounding accuracy $Acc@0.5$, which indicates whether the tIoU between the grounding result generated by a model and the ground truth is higher than 0.5.

| Method | DenseCaption | Charades-STA |
|---|---|---|
| | $Acc@0.5$ | $Acc@0.5$ |
| RANDOM | 18.1% | 14.0% |
| FIXED | 25.4% | 23.1% |
| TALL (Gao et al.) | 26.9% | 25.8% |
| MCN (Hendricks et al.) | 27.4% | 30.8% |
| RL-Loc (Yeung et al.) | 34.7% | 32.5% |
| Ours | **36.9%** | **36.7%** |

Table 1: The evaluation results of different methods.

### 4.2 Implementation Details

In the training phase, Adam with initial learning rate of 1e-3 is used for optimization. In the test phase, an action is greedily sampled at each time step: $a^{(t)} = \arg\max_{a_i^{(t)}} \pi(a_i^{(t)}|s^{(t)}, \theta)$. The initial temporal grounding location is set to the central half of the entire video, namely, $L^{(0)} = [N/4, 3N/4]$, where $N$ is the total length of the video. Because no optical flow frames are provided in the DenseCaption dataset, we use only the RGB frames to train our models. However, Charades-STA does not have such limitation, thus all the models trained on it use both optical flow and RGB modalities.

The parameters used in the experiments are set as follows. $T_{max}$ is set to 10. The natural language description is embedded into a 2400-dimension vector via skip-thought (Kiros et al. 2015). In the observation network, the output size of the FC layer for encoding the description feature is set to 1024. Both global feature and local feature are encoded into 512-dimension vectors by FC layers. The normalized temporal boundary $L^{(t-1)}$ is embedded into a 128-dimension vector with a FC layer. The dimension of the state vector $s^{(t)}$ at the $t^{th}$ step is 1024. The GRU cell which is used to model the sequential decision process gets hidden size of 1024. $\phi$ is set to 0.001. $\gamma$ is set to 0.3 on DenseCaption and 0.4 on Charades-STA.

### 4.3 Comparison with Baselines

Action localization methods can be adapted to this task by feeding video and sentence features and then making 0 vs. 1 action localization. In fact, both Gao et al. (2017) (denoted by TALL) and Hendricks et al. (2017) (denoted by MCN) have adopted this idea. We use both methods as baselines and implement them using the source codes released by their authors. Yeung et al. (2016) (denoted by RL-Loc) uses RL for action localization, we also adapt it for video grounding and use it as a baseline. In addition, to better evaluate our proposed model, we also use two hand-crafted approaches as baselines. First, a random strategy (denoted by RANDOM) is employed by randomly selecting a video clip with half of the entire video length for arbitrary descriptions. Second, a fixed strategy (denoted by FIXED) is employed by selecting the central part with a fixed length ($N/2$ for DenseCaption and $N/3$ for Charades-STA) for any descriptions.

Table 1 shows the evaluation results. We observe that: 1) our method achieves the best performance in comparison with all baselines on both datasets; 2) our method sig-

| $\phi$ | DenseCaption |  | Charades-STA |  |
|---|---|---|---|---|
|  | # Steps | $Acc@0.5$ | # Steps | $Acc@0.5$ |
| 1 | 1.00 | 25.4% | 1.0 | 17.9% |
| 0.1 | 5.30 | 36.1% | 5.17 | 36.3% |
| 0.01 | 9.96 | 36.7% | 9.97 | **36.7%** |
| 0.001 | 9.99 | **36.9%** | 9.99 | **36.7%** |

Table 2: $Acc@0.5$ and average number of glimpses evaluated with different $\phi$ settings.

nificantly outperforms previous state-of-the-art methods by substantial margins: the accuracy is improved by 10.0% and 10.9% over TALL; 9.5% and 5.9% over MCN; 2.2% and 4.2% over RL-Loc on DenseCaption and Charades-STA, respectively. The evaluation results demonstrate that our model achieves a significant improvement over all previous state-of-the-art models. This suggests that it is a more natural formulation of the task by adopting the read, watch, and move strategy with sequential decision making. As for the inference computation cost, both our model and RL-Loc only need to observe up to 10 clips per video. MCN splits a video into 6 segments and needs to enumerate all 21 possible video clips. TALL adopts sliding windows of 128 and 256 frames with an overlap ratio of 0.8 and needs to examine 45.8 candidates per video on average.

## 4.4 Ablation Study

In this section, we perform a series of ablation studies to understand the importance of different factors, reward, and components. To this end, we train multiple variations of our model with different settings or configurations. In the following experiments, we use RGB feature of the videos on DenseCaption while both RGB and flow features on Charades-STA. Specifically, when two modalities are used, both the global feature $V_G$ and the local feature $V_L^{(t-1)}$ are the concatenation of RGB and optical flow features.

**Impact of Penalty Factors** In the reward function, we have proposed to punish the agent for taking too many steps to strike a desirable trade-off between grounding accuracy and computation cost. In this experiment, we examine the impact of the parameter $\phi$ while keeping $\gamma$ fixed. Table 2 shows the results of $Acc@0.5$ and the average number of steps the agent takes. From the results, we can clearly observe that the average number of steps gets higher when $\phi$ gets lower on both datasets, and the grounding accuracy increases when $\phi$ decreases. Specially, when the penalty factor was set to 1, the average number of steps is 1, which indicates that the agent takes the STOP action at the first step. On the contrary, if $\phi$ was set to 0.001, the average number of steps is 9.99, which indicates that the agent has fully utilized the possible exploring steps (which is up-bounded by $T_{max} = 10$) to adjust the temporal grounding boundaries, thus the best accuracy is achieved. This shows that $\phi$ can be utilized to strike a desirable trade-off between the accuracy and computation cost. For example, setting $\phi$ to 0.1 could be a balanced choice if we would like to achieve better grounding accuracy in fewer steps.

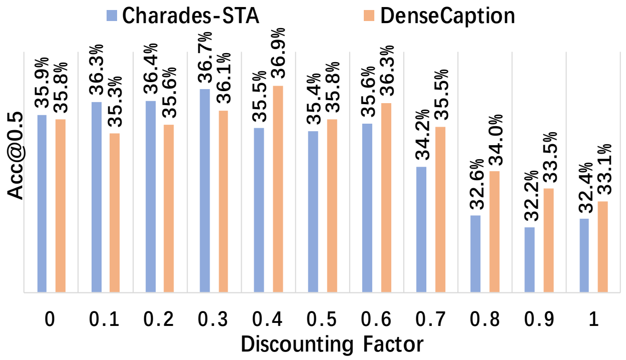

Figure 3: Results of the methods with different discount factor $\gamma$ settings on Charades-STA and DenseCaption.

**Impact of Accumulated Reward** In order to build connections inside the action sequence, we propose to trace reward of the current step back to the previous steps, so that the reward can be accumulated. In this experiment, we fix $\phi$ to 0.001 and evaluate the impact of the discount factor $\gamma$ in our final reward $R_t$. We range $\gamma$ from 0 to 1 with a step size of 0.1. Then, we train multiple variations of our model with different discounting factors, and examine their impact. Figure 3 shows the experimental results. From the results, it is observed that the grounding accuracy obtained with either 0 or 1 of $\gamma$ is lower than most of those obtained with the values of $\gamma$ in between 0 and 1. Intuitively, when $\gamma$ is set to 0, it indicates that the reward of the current action has no impact on its follow-up actions. This is not a plausible reward strategy for the entire sequential decision making process because the dependency between successive actions is not modeled. Conversely, when $\gamma$ is set to 1, it indicates that the reward of the current action largely depends on its successive actions. This is also not a plausible reward strategy because even if the $t^{th}$ action is an unsuccessful attempt, the accumulated reward $R_t$ can be still positive if there is a chance that any of its successive actions get a positive reward. In this experiment, we empirically observed that when $\gamma$ was set to 0.3 on Charades-STA and 0.4 on DenseCaption, the model achieves the best grounding accuracy on each dataset.

**Importance of Different Components** Here we investigate the effectiveness of each component in our proposed model by conducting an ablation study. In the experiments, $\phi$ was set to 0.001 to enable the agent to fully explore the grounding result within the maximum steps allowed, while $\gamma$ was set to 0.3 on Charades-STA and 0.4 on DenseCaption.

We first study the effectiveness of the introduced location feature. To this end, we train a new model by removing this feature from the observation network, such that only $V_G$, $E$ and $V_L^{(t-1)}$ are encoded at the $t^{th}$ step. Then we investigate the importance of the proposed multi-task learning, which helps the agent to better capture the representation of the current state of the environment. To accomplish this, we train multiple variations of our model by disabling the two modules (tIoU regression and location regression) one after the other while keeping other settings exactly as what they are.

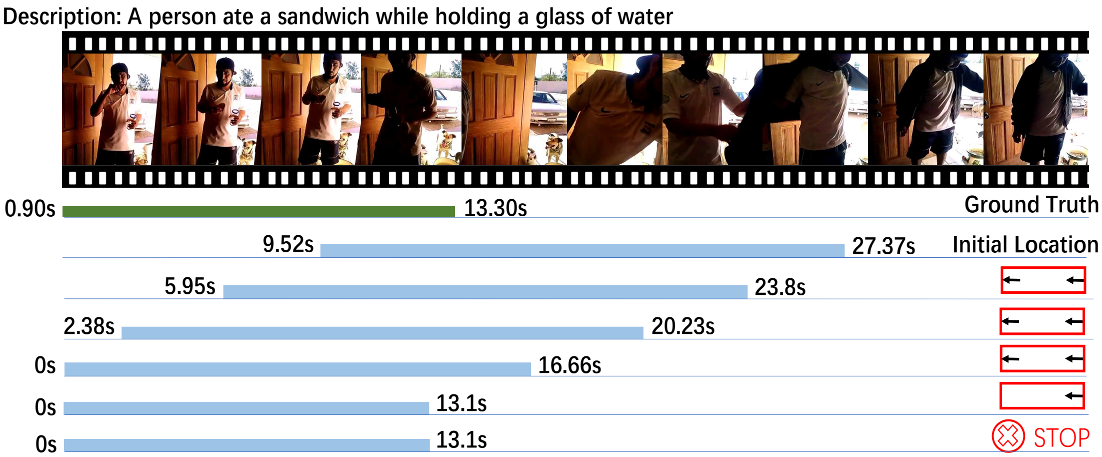

Figure 4: Illustration of the action sequence executed when our model grounds the given description in a video.

We conduct experiments on both datasets and the results are detailed in Table 3. From the results, we make the following observations.

First, the performance significantly drops on both datasets after disabling all three components, which particularly results in a dramatic decrease of $Acc@0.5$ by 23.7% (from 36.7% to 13.0%) on Charades-STA. The leave-one-out analysis also reveals that the performance drops by a large margin after removing one or more components. This demonstrates that all the three components can significantly contribute to the grounding effectiveness.

Second, we compare the models with or without the location feature used as a component of the environment. The results reveal that the grounding accuracy drops 3.4% and 1.9% in terms of $Acc@0.5$ on DenseCaption and Charades-STA, respectively. This verifies that explicit location feature is important to help model the environment for the observation network.

Third, the results show that $Acc@0.5$ decreases from 36.9% to 35.9% on DenseCaption and from 36.7% to 35.1% on Charades-STA after disabling the component of location regression. Moreover, $Acc@0.5$ further reduces to 34.5% on DenseCaption and 34.9% on Charades-STA after disabling both the two components of tIoU regression and location regression. This demonstrates that the overall grounding performance will degrade considerably without the consideration of supervised learning. This also confirms the importance of modeling the environment by combining supervised learning with reinforcement learning in a multi-task learning framework, which can help the agent obtain more accurate information about the environment and better explore the state space to encounter more reward.

| Configurations | | | $Acc@0.5$ | |
|---|---|---|---|---|
| Loc | tIoU-R | Loc-R | DenseCaption | Charades-STA |
| × | × | × | 31.5% | 13.0% |
| × | √ | √ | 33.5% | 34.8% |
| √ | √ | × | 35.9% | 35.1% |
| √ | × | × | 34.5% | 34.9% |
| √ | √ | √ | **36.9%** | **36.7%** |

Table 3: Results with different network configurations. Loc refers to "Explicit Location Feature", tIoU-R refers to "tIoU Regression", and Loc-R refers to "Location Regression".

**Multi-Modality Experiment** We investigate the effect of combining both RGB and flow features on Charades-STA as a case study. The experimental results show that the grounding accuracies achieved by the models trained with RGB feature and flow feature are 36.2% and 35.8%, respectively. However, the grounding accuracy is slightly increased to 36.7% when the two modalities are fused by feature concatenation. The performance could be further improved with better fusion methods, which we leave as a future work.

### 4.5 Qualitative Results

To visualize how the agent adjusts grounding boundaries step by step, we illustrate an action sequence executed by our agent in the procedure of grounding a natural language description in a video (see Figure 4). The agent takes five steps from the initial location and finally grounds the description correctly in the video.

## 5 Conclusion

In this paper, we study the problem of temporally grounding natural language descriptions in videos. This task can be formulated as a sequential decision making process and it fits naturally into the reinforcement learning paradigm. Therefore, we model this task as controlling an agent to read the description, to watch the video as well as the current localization, and then to move the temporal grounding boundaries iteratively to find the best matching clip. Combined with supervised learning in a multi-task learning framework, our model can be easily trained end-to-end. Experiments demonstrate that our method achieves a new state-of-the-art performance on two well-known datasets.

## References

Bojanowski, P.; Lajugie, R.; Grave, E.; Bach, F.; Laptev, I.; Ponce, J.; and Schmid, C. 2015. Weakly-supervised alignment of video with text. In *ICCV*, 4462–4470.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Schwenk, H.; Schwenk, H.; Bengio, Y.; Bengio, Y.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, 1724–1734.

Endo, K.; Aono, M.; Nichols, E.; and Funakoshi, K. 2017. An attention-based regression model for grounding textual phrases in images. In *IJCAI*, 3995–4001.

Gan, Z.; Gan, C.; He, X.; Pu, Y.; Tran, K.; Gao, J.; Carin, L.; and Deng, L. 2017. Semantic compositional networks for visual captioning. In *CVPR*, 1141–1150.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. TALL: Temporal activity localization via language query. In *ICCV*, 5267–5275.

Gao, J.; Sun, C.; and Nevatia, R. 2016. ACD: Action concept discovery from image-sentence corpora. In *ICMR*, 31–38.

Han, J.; Yang, L.; Zhang, D.; Chang, X.; and Liang, X. 2018. Reinforcement cutting-agent learning for video object segmentation. In *CVPR*, 9080–9089.

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *ICCV*, 5803–5812.

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *NIPS*, 3294–3302.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Niebles, J. C. 2017. Dense-captioning events in videos. In *ICCV*, 706–715.

Li, D.; Wu, H.; Zhang, J.; and Huang, K. 2018. A2-RL: Aesthetics aware reinforcement learning for image cropping. In *CVPR*, 8193–8201.

Liang, X.; Hu, Z.; Zhang, H.; Gan, C.; and Xing, E. P. 2017. Recurrent topic-transition gan for visual paragraph generation. In *ICCV*, 3362–3371.

Lin, D.; Fidler, S.; Kong, C.; and Urtasun, R. 2014. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*, 2657–2664.

Lin, T.; Zhao, X.; and Shou, Z. 2017. Single shot temporal action detection. In *ACM MM*, 988–996.

Liu, X.; Wang, J.; Wen, S.; Ding, E.; and Lin, Y. 2017. Localizing by describing: Attribute-guided attention localization for fine-grained recognition. In *AAAI*, 4190–4196.

Liu, M.; Wang, X.; Nie, L.; He, X.; Chen, B.; and Chua, T.-S. 2018. Attentive moment retrieval in videos. In *SIGIR*, 15–24.

Mnih, V.; Heess, N.; Graves, A.; et al. 2014. Recurrent models of visual attention. In *NIPS*, 2204–2212.

Pan, Y.; Mei, T.; Yao, T.; Li, H.; and Rui, Y. 2016. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 4594–4602.

Pirinen, A., and Sminchisescu, C. 2018. Deep reinforcement learning of region proposal networks for object detection. In *CVPR*, 6945–6954.

Plummer, B. A.; Wang, L.; Cervantes, C. M.; Caicedo, J. C.; Hockenmaier, J.; and Lazebnik, S. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2641–2649.

Regneri, M.; Rohrbach, M.; Wetzel, D.; Thater, S.; Schiele, B.; and Pinkal, M. 2013. Grounding action descriptions in videos. *Transactions of the Association of Computational Linguistics* 1:25–36.

Rohrbach, A.; Rohrbach, M.; Hu, R.; Darrell, T.; and Schiele, B. 2016. Grounding of textual phrases in images by reconstruction. In *ECCV*, 817–834.

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S.-F. 2017. CDC: Convolutional-De-Convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 5734–5743.

Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 510–526.

Singh, B.; Marks, T. K.; Jones, M.; Tuzel, O.; and Shao, M. 2016. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *CVPR*, 1961–1970.

Sutton, R. S., and Barto, A. G. 2011. *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Tellex, S., and Roy, D. 2009. Towards surveillance video search by natural language query. In *CIVR*, 38:1–38:8.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 4489–4497.

Wang, X.; Chen, W.; Wu, J.; Wang, Y.-F.; and Yang Wang, W. 2018. Video captioning via hierarchical reinforcement learning. In *CVPR*, 4213–4222.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Xiao, F.; Sigal, L.; and Jae Lee, Y. 2017. Weakly-supervised visual grounding of phrases with linguistic structures. In *CVPR*, 5945–5954.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015a. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2048–2057.

Xu, R.; Xiong, C.; Chen, W.; and Corso, J. J. 2015b. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2346–2352.

Yeung, S.; Russakovsky, O.; Mori, G.; and Fei-Fei, L. 2016. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2678–2687.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In *ICCV*, 2914–2923.