Tree-Structured Policy based Progressive Reinforcement Learning for Temporally Language Grounding in Video

Jie Wu[1], **Guanbin Li**[1*], **Si Liu**[2], **Liang Lin**[1,3]
[1] Sun Yat-sen University [2] Beihang University, [3] DarkMatter AI Research.
wujie23@mail2.sysu.edu.cn, liguanbin@mail.sysu.edu.cn, liusi@buaa.edu.cn, linliang@ieee.org

Abstract

Temporally language grounding in untrimmed videos is a newly-raised task in video understanding. Most of the existing methods suffer from inferior efficiency, lacking interpretability, and deviating from the human perception mechanism. Inspired by human's coarse-to-fine decision-making paradigm, we formulate a novel Tree-Structured Policy based Progressive Reinforcement Learning (TSP-PRL) framework to sequentially regulate the temporal boundary by an iterative refinement process. The semantic concepts are explicitly represented as the branches in the policy, which contributes to efficiently decomposing complex policies into an interpretable primitive action. Progressive reinforcement learning provides correct credit assignment via two task-oriented rewards that encourage mutual promotion within the tree-structured policy. We extensively evaluate TSP-PRL on the Charades-STA and ActivityNet datasets, and experimental results show that TSP-PRL achieves competitive performance over existing state-of-the-art methods.

Introduction

We focus on the task of temporally language grounding in a video, whose goal is to determine the temporal boundary of the segments in the untrimmed video that corresponds to the given sentence statement. Most of the existing competitive approaches (Anne Hendricks et al. 2017; Gao et al. 2017; Liu et al. 2018; Ge et al. 2019; Xu et al. 2019) are based on extensive temporal sliding windows to slide over the entire video or rank all possible clip-sentence pairs to obtain the grounding results. However, these sliding window based methods suffer from inferior efficiency and deviate from the human perception mechanism. When humans locate an interval window associated with a sentence description in a video, they tend to assume an initial temporal interval first, and achieve precise time boundary localization through cross-modal semantic matching analysis and sequential boundary adjustment (e.g., scaling or shifting).

Looking deep into human's thinking paradigm (Mancas et al. 2016), people usually deduce a coarse-to-fine deliberation process to render a more reasonable and interpretable decision in daily life. Namely, people will first roughly determine the selection range before making a decision, then choose the best one among the coarse alternatives. This top-down coarse-to-fine deliberation has been explored in the task of machine translation, text summarization and so on (Xia et al. 2017). Intuitively, embedding this mode of thinking into our task can efficiently decompose complex action policies, reduce the number of search steps while increasing the search space, and obtain more impressive results in a more reasonable way. To this end, we formulate a Tree-Structured Policy based Progressive Reinforcement Learning framework (TSP-PRL) to imitate human's coarse-to-fine decision-making scheme. The tree-structured policy in TSP-PRL consists of root policy and leaf policy, which respectively correspond to the process of coarse and fine decision-making stage. And a more reasonable primitive action is proposed via these two-stages selection. The primitive actions are divided into five classes related to semantic concepts according to the moving distance and directions: scale variation, markedly left shift, markedly right shift, marginally left adjustment and marginally right adjustment. The above semantic concepts are explicitly represented as the branches into the tree-structured policy, which contributes to efficiently decomposing complex policies into an interpretable primitive action. In the reasoning stage, the root policy first roughly estimates the high-level semantic branch that can reduce the semantic gap to the most extent. Then the leaf policy reasons a refined primitive action based on the selected branch to optimize the boundary. We depict an example of how TSP-PRL addresses the task in Figure 1. As can be seen in the figure, the agent first markedly right shift the boundary to eliminate the semantic gap. Then it resorts to scale contraction and marginally adjustment to obtain an accurate boundary.

The tree-structured policy is optimized via progressive re-

*Corresponding author is Guanbin Li. This work was supported in part by the National Key Research and Development Program of China under Grant No. 2018YFC0830103, in part by the National Natural Science Foundation of China under Grant No.61976250 and No.61876177, in part by the National High Level Talents Special Support Plan (Ten Thousand Talents Program). This work was also supported by sponsored by CCF-Tencent Open Research Fund (CCF-Tencent IAGR20190106).

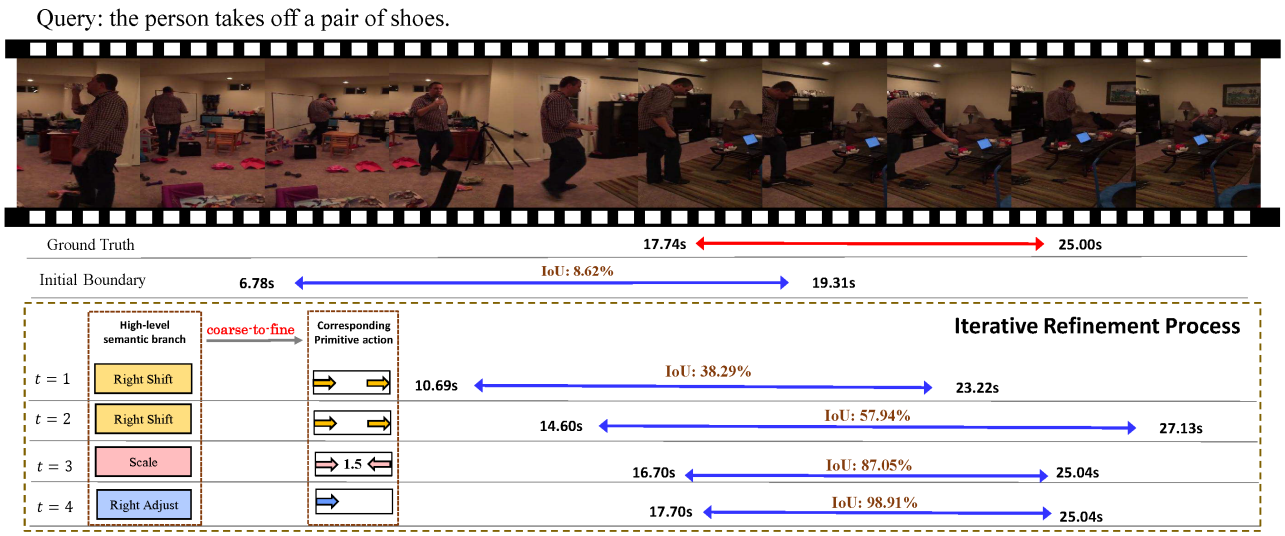


Figure 1: An example showing how TSP-PRL addresses the task in an iterative refinement manner. A more interpretable primitive action is proposed by the tree-structured policy, which consists of root policy and leaf policy to imitate human's coarse-to-fine decision-making scheme.

inforcement learning, which determines the selected single policy (root policy or leaf policy) in the current iteration while stabilizing the training process. The task-oriented reward settings in PRL manages to provide correct credit assignment and optimize the root policy and leaf policy mutually and progressively. Concretely, the external environment provides rewards for each leaf strategy and the root strategy does not interact directly with the environment. PRL measures the reward for the root policy from two items: 1) the intrinsic reward for the selection of high-level semantic branch; 2) the extrinsic reward that reflects how the subsequent action executed by the selected semantic branch influences the environment.

Extensive experiments on Charades-STA (Sigurdsson et al. 2016; Gao et al. 2017) and ActivityNet (Krishna et al. 2017) datasets prove that TSP-PRL achieves competitive performance over existing leading and baseline methods on both datasets. The experimental results also demonstrate that the proposed approach can (i) efficiently improve the ability to discover complex policies which can hardly be learned by flat policy; (ii) provide more comprehensive assessment and appropriate credit assignment to optimize the tree-structured policy progressively; and (iii) determine a more accurate stop signal at an iterative process. The source code as well as the trained models have been released at *https://github.com/WuJie1010/TSP-PRL*.

Related work

Temporally Language Grounding in Video. Temporally language grounding in the video is a challenging task which requires both language and video understanding and needs to model the fine-grained interactions between the verbal and visual modalities. Gao *et al.* (Gao et al. 2017) proposed a cross-modal temporal regression localizer (CTRL) to jointly model language query and video clips, which adopts sliding windows over the entire video to obtain the grounding results. Hendricks *et al.*(Anne Hendricks et al. 2017) designed a moment context network (MCN) to measure the distance between visual features and sentence embedding in a shared space, ranking all possible clip-sentence pairs to locate the best segments. However, the above approaches are either inefficient or inflexible since they carry out overlapping sliding window matching or exhaustive search. Chen *et al.* (Chen et al. 2018a) designed a dynamic single-stream deep architecture to incorporate the evolving fine-grained frame-by-word interactions across video-sentence modalities. This model performs efficiently, which only needs to process the video sequence in one single pass. Zhang *et al.* (Zhang et al. 2019) exploited graph-structured moment relations to model temporal structures and improve moment representation explicitly. He *et al.* (He et al. 2019) first introduced the reinforcement learning paradigm into this task and treated it as a sequential decision-making task. Inspired by human's coarse-to-fine decision-making paradigm, we construct a tree-structured policy to reason a series of interpretable actions and regulate the boundary in an iterative refinement manner.

Reinforcement Learning. Recently, reinforcement learning (RL) technique (Williams 1992) has been successfully popularized to learn task-specific policies in various image/video-based AI tasks. These tasks can be generally formulated as a sequential process that executes a series of actions to finish the corresponding objective. In the task of multi-label image recognition, Chen *et al.* (Chen et al. 2018b) proposed a recurrent attentional reinforcement learning method to iteratively discover a sequence of attentional and informative regions. Shi *et al.*(Shi et al. 2019) implemented deep reinforcement learning and developed a novel attention-aware face hallucination framework to generate a high-resolution

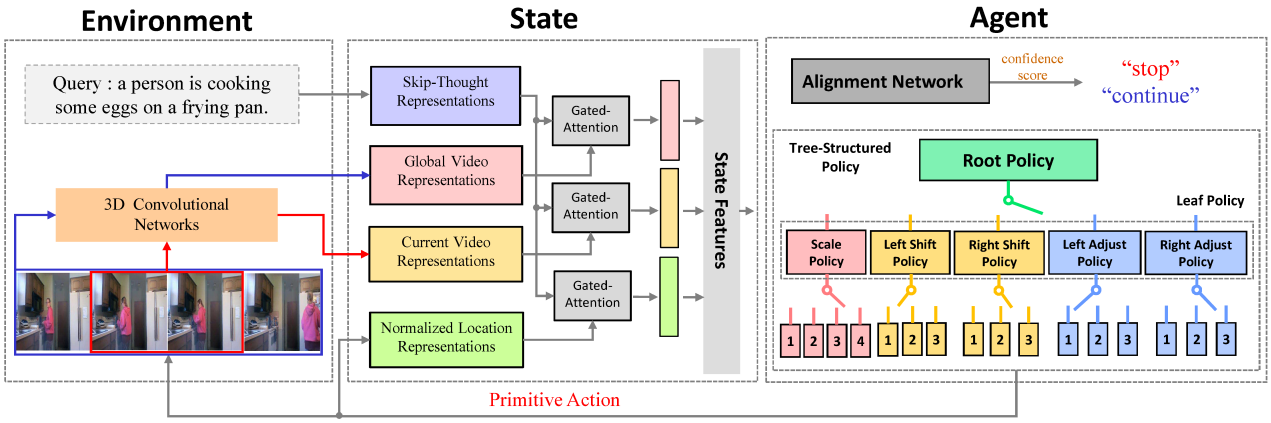


Figure 2: The overall pipeline of the proposed TSP-PRL framework. The agent receives the state from the environment (video clips) and estimates a primitive action via tree-structured policy. The action selection is depicted by a switch $\multimap$ over the interface $\perp$ in the tree-structured policy. The alignment network will predict a confidence score to determine when to stop.

face image from a low-resolution input. Wu *et al.* (Wu et al. 2019a) designed a new content sensitive and global discriminative reward function to encourage generating more concrete and discriminative image descriptions. In the video domain, RL has been widely used in temporal action localization (Yeung et al. 2016) and video recognition (Wu et al. 2019b). In this paper, we design a progressive RL approach to train the tree-structured policy, and the task-oriented reward settings contribute to optimizing the root policy and leaf policy mutually and stably.

The Proposed Approach

Markov Decision Process Formulation

In this work, we cast the temporally language grounding task as a Markov Decision Process (MDP), which is represented by states $s \in \mathcal{S}$, action tuple $\langle a^r, a^l \rangle$, and transition function $\mathcal{T} : (s, \langle a^r, a^l \rangle) \rightarrow s'$. a^r and a^l denote the action proposed by root policy and leaf policy, respectively. The overall pipeline of the proposed Tree-Structured Policy based Progressive Reinforcement Learning (TSP-PRL) framework is depicted in Figure 2.

State. A video is firstly decomposed into consecutive video units (Gao et al. 2017) and each video unit is used to extract unit-level feature through the feature extractor φ_v (Tran et al. 2015; Wang et al. 2016). Then the model resorts to uniformly sampling strategy to extract ten unit-level features from the entire video, which are concatenated as the global video representation V^g. For the sentence query L, the skip-thought encoder φ_s (Kiros et al. 2015) is utilized to generate the language representation $E = \varphi_s(L)$. When the agent interacts with the environment, the above features are retained. At each time step, the action executed by the agent will change the boundary and obtain a refined video clip. The model samples ten unit-level features inside the boundary and concatenate these features as the current video feature V^c_{t-1}, $t = 1, 2, ..., T_{max}$. We explicitly involve the normalized boundary $L_{t-1} = [l^s_{t-1}, l^e_{t-1}]$ into the state feature (He et al. 2019), where l^s_{t-1} and l^e_{t-1} denote the start point and end point respectively. Then the gated-attention (Chaplot et al. 2018) mechanism is applied to gain multi-modal fusion representation of verbal and visual modalities:

$$A^{EG}_t = \sigma(E) \odot V^g, \quad A^{EC}_t = \sigma(E) \odot V^c_{t-1}, \quad A^{EL}_t = \sigma(E) \odot L_{t-1}, \tag{1}$$

where σ denotes the sigmoid activation function and $\odot$ is the Hadamard product. The above gated attention features are concatenated and fed into a fully-connected layer ϕ to obtain the state representation s_t:

$$s_t = \phi(A^{EG}_t, A^{EC}_t, A^{EL}_t) \tag{2}$$

An additional GRU (Cho et al. 2014) layer is adopted to process the state features before feeding them into the tree-structured policy, which manages to develop high-level temporal abstractions and lead to a more generalizable model.

Hierarchical Action Space. In our work, the boundary movement is based on the clip-level and each boundary consists of a series of video clips. All primitive actions can be divided into five classes related to semantic concepts according to the moving distance and directions, which results in a hierarchical action space on the whole. These semantic concepts are explicitly represented as the branches into the tree-structured policy, resulting in five high-level semantic branches to contain all primitive actions in this task: scale variation, markedly left shift, markedly right shift, marginally left adjustment and marginally right adjustment. i) The scale variation branch contains four primitive actions: extending/shortening ξ times w.r.t center point. ξ is set to 1.2 or 1.5; 2) Three actions are included in the markedly left shift branch: shifting start point/end point/start & end point backward ν. ν is fixed to $N/10$, where N denotes the number of the clip of the entire video; 3) The actions in the markedly right shift branch is symmetry with the markedly left shift: shifting start point/end point/start & end point forward ν; 4) Except for the moving scale, the actions in the marginally left adjustment branch is similar to the markedly left shift branch: shifting start point/end point/start & end

point backward Z frame; The size of Z is constrained by the video lengths; 5) The marginally right adjustment branch also involves three primitive actions: shifting start point/end point/start & end point forward Z frame.

Tree-Structured Policy. One of our key ideas is that the agent needs to understand the environmental state well and reason a more interpretable primitive action. Inspired by human's coarse-to-fine decision-making paradigm, we design a tree-structured policy to decompose complex action policies and propose a more reasonable primitive action via two-stages selection, instead of using a flat policy that maps the state feature to action directly (He et al. 2019). As shown in the right half of Figure 2, the tree-structured policy consists of a root policy and a leaf policy at each time step. The root policy $\pi^r(a_t^r|s_t)$ decides which semantic branch will be primarily relied on. The leaf policy $\pi^l(a_t^l|s_t, a_t^r)$ consists of five sub-policies, which corresponds to five high-level semantic branches. The selected semantic branch will reason a refined primitive action via the corresponding sub-policy. The root policy aims to learn to invoke the correct sub-policy from the leaf policy in the following different situations: (1) The scaling policy should be selected when the scale of predicted boundary is quite mismatched with the ground-truth boundary; (2) When the predicted boundary is far from the ground-truth boundary, the agent should execute the left or right shift policy; (3) The primitive action should be sampled from the left or right adjust policy when most of the two boundaries intersect but with some deviation. At each time step, the tree-structured policy first samples a_t^r from root policy π^r to decide the semantic branch:

$$a_t^r \sim \pi^r(a_t^r|s_t). \tag{3}$$

And a primitive action is sampled from the leaf policy π^l related to the selected semantic branch:

$$a_t^l \sim \pi^l(a_t^l|s_t, a_t^r). \tag{4}$$

Tree-Structured Policy based Progressive Reinforcement Learning

Rewards. Temporal IoU is adopted to measure the alignment degree between the predicted boundary $[l^s, l^e]$ and ground-truth boundary $[g^s, g^e]$:

$$U_t = \frac{\min(g^e, l_t^e) - \max(g^s, l_t^s)}{\max(g_e, l_t^e) - \min(g^s, l_t^s)}. \tag{5}$$

The reward setting for this task should provide correct credit assignment, encouraging the agent to take fewer steps to obtain accurate grounding results. We define two task-oriented reward functions to select an accurate high-level semantic branch and the corresponding primitive action, respectively. The first reward r_t^l is the leaf reward, which reveals the influence of the primitive actions a_t^l to the current environment. It can be directly obtained in the environment through temporal IoU. We explicitly provide higher leaf reward when the primitive action attempts to obtain better grounding results and the temporal IoU is higher than 0.5:

$$r_t^l = \begin{cases} \zeta + U_t & U_t > U_{t-1}; U_t > 0.5 \\ \zeta & U_t > U_{t-1}; U_t \leq 0.5 \\ -\zeta/10 & U_{t-1} \geq U_t \geq 0 \\ -\zeta & \text{otherwise} \end{cases}, \tag{6}$$

where ζ is a factor that determines the degree of reward.

The second reward is the root reward r_t^r, which should be determined deliberately since the action executed by root policy does not interact with the environment directly. To provide comprehensive assessment and correct credit assignment, r_t^r is defined to include two items: 1) the intrinsic reward term that represents the direct impact of a_t^r for semantic branch selection and 2) the extrinsic reward term reflects the indirect influence of the subsequent primitive action executed by the selected branch for the environment. In order to estimate how well the root policy chooses the high-level semantic branch, the model traverses through all possible branches and reasons the corresponding primitive actions to the environment, which results in five different IoU. The max IoU among these five IoU is defined as $U_t^{\max}$. Then the root reward r_t^r is designed as follow:

$$r_t^r = \begin{cases} \underbrace{\zeta}_{\text{intrinsic reward item}} + \underbrace{U_t - U_{t-1}}_{\text{extrinsic reward item}} & U_t = U_t^{\max} \\ \underbrace{U_t - U_t^{\max}}_{\text{intrinsic reward item}} + \underbrace{U_t - U_{t-1}}_{\text{extrinsic reward item}} & \text{otherwise} \end{cases}, \tag{7}$$

where U_0 denotes the temporal IoU between initial boundary and the ground-truth boundary. The diagram of how the root reward and leaf reward are obtained in the framework is depicted in Figure 3.

Progressive Reinforcement Learning. Progressive Reinforcement Learning (PRL) is designed on the basis of the advantage actor-critic (A2C) (Sutton and Barto 2018) algorithm to optimize the overall framework. Policy function $\pi^r(a_t^r|s_t)$ and $\pi^l(a_t^l|s_t, a_t^r)$ estimate the probability distribution over possible actions in the corresponding action space. These two policies are separate and each is equipped with a value approximator $V^r(s_t)$ and $V^l(s_t, a_t^r)$, which is designed to compute a scalar estimate of reward for the corresponding policy.

Starting from the initial boundary, the agent invokes the tree-structured policy iteratively in the interaction process. We depict how the tree-structured policy works iteratively in Figure 3. From the figure, we can observe that the agent samples actions from root policy and leaf policy consecutively at each time step. The action will trigger a new state, which is fed into the tree-structured policy to execute the next actions. Given a trajectory in an episode $\Gamma = \{\langle s_t, \pi^r(\cdot|s_t), a_t^r, r_t^r, \pi^l(\cdot|s_t, a_t^r), a_t^l, r_t^l\rangle, t = \{1, \cdots, T_{max}\}\}$, PRL algorithm maximizes the objective of root policy $\mathcal{L}_{root}(\theta_{\pi^r})$ and leaf policy $\mathcal{L}_{leaf}(\theta_{\pi^l})$:

$$\begin{aligned} \mathcal{L}_{root}(\theta_{\pi^r}) = -\frac{1}{M}\sum_{m=1}^{M}\sum_{t=1}^{T_{max}} [&\log\pi^r(a_t^r|s_t)(R_t^r - V^r(s_t)) \\ &+ \alpha H(\pi^r(a_t^r|s_t))], \end{aligned} \tag{8}$$

$$\begin{aligned} \mathcal{L}_{leaf}(\theta_{\pi^l}) = -\frac{1}{M}\sum_{m=1}^{M}\sum_{t=1}^{T_{max}} [&\log\pi^l(a_t^l|s_t, a_t^r)(R_t^l - V^l(s_t, a_t^r)) \\ &+ \alpha H(\pi^l(a_t^l|s_t, a_t^r))], \end{aligned} \tag{9}$$

where M denotes the size of a mini-batch and T_{max} is the max time step in an episode. $R_t^r - V^r(s_t)$ and $R_t^l -$

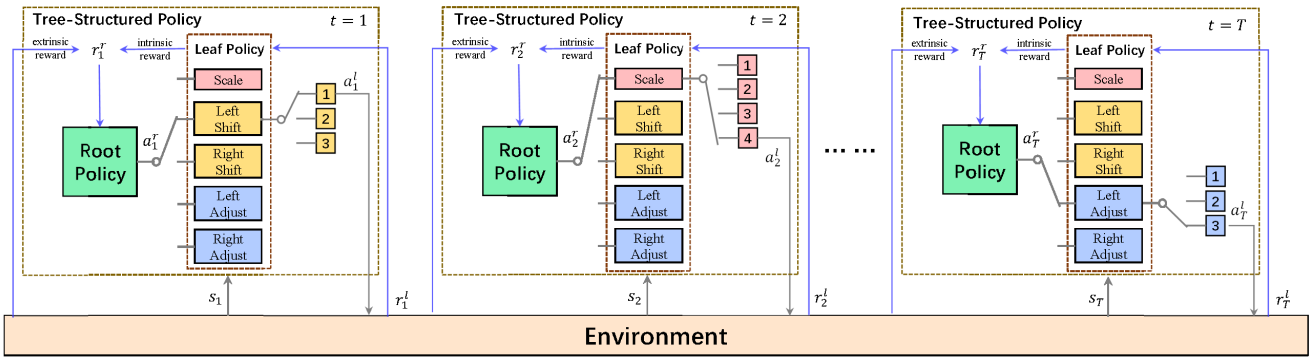


Figure 3: An illustration of how the tree-structured policy works iteratively. The solid blue line represents how the root reward and leaf reward are obtained from the proposed framework.

$V^l(s_t, a_t^r)$ denote the advantage functions in the A2C setting. $H()$ is the entropy of policy networks and the hyper-parameters α controls the strength of entropy regularization term, which is introduced to increase the diversity of actions. θ_{π^r} and θ_{π^l} are the parameters of the policy networks. Here, the model only back-propagates the gradient for the selected sub-policy in leaf policy. The reward of the following-up steps should be traced back to the current step since it is a sequential decision-making problem. The accumulated root reward function R_t^r is computed as follows:

$$R_t^r = \begin{cases} r_t^r + \gamma V^r(s_t) & t = T_{max} \\ r_t^r + \gamma R_{t+1}^r & t = 1, 2, ..., T_{max} - 1 \end{cases}, \tag{10}$$

where γ is a constant discount factor and the accumulated leaf reward R_t^l is obtained in a similar way. In order to optimize the value network to provide an estimation of the expected sum of rewards, we minimize the squared difference between the accumulated reward and the estimated value, and minimize the value loss:

$$\begin{aligned} \mathcal{L}_{root}(\theta_{V^r}) &= \frac{1}{M}\sum_{m=1}^{M}\sum_{t=1}^{T_{max}}(R_t^r - V^r(s_t))^2, \\ \mathcal{L}_{leaf}(\theta_{V^l}) &= \frac{1}{M}\sum_{m=1}^{M}\sum_{t=1}^{T_{max}}(R_t^l - V^l(s_t, a_t^r))^2 \end{aligned} \tag{11}$$

where θ_{V^r} and θ_{V^l} are the parameters of the value networks.

Optimizing the root and leaf policies will simultaneously lead to the unstable training procedure. To avoid this, we design a progressive reinforcement learning (PRL) optimization procedure: for each set of K iterations, PRL keeps one policy fixed and only trains the other policy. When reaching K iterations, it switches the policy that is trained. The tree-structured policy based progressive reinforcement learning can be summarized as:

$$\psi = \lfloor \frac{i}{K} \rfloor \bmod 2, \tag{12}$$

$$\begin{aligned} \mathcal{L}_{tree} =& \psi \times [\mathcal{L}_{root}(\theta_{\pi^r}) + \mathcal{L}_{root}(\theta_{V^r})] \\ &+ (1-\psi) \times [\mathcal{L}_{leaf}(\theta_{\pi^l}) + \mathcal{L}_{leaf}(\theta_{V^l})], \end{aligned} \tag{13}$$

where ψ is a binary variable indicating the selection of the training policy. i denotes the number of iterations in the entire training process. $\lfloor \rfloor$ is the lower bound integer of the division operation and $\bmod$ is the modulo function. These two policies promote each other mutually, as leaf policy provides accurate intrinsic rewards for root policy while the root policy selects the appropriate high-level semantic branch for further refinement of the leaf policy. The better leaf policy is, the more accurate intrinsic rewards will be provided. The more accurate the upper branch policy is selected, the better the leaf policy can be optimized. This progressive optimization ensures the agent to obtain a stable and outstanding performance in the RL setting. During testing, the tree-structured policy takes the best actions tuple $\langle a^r, a^l \rangle$ at each time step iteratively to obtain the final boundary.

Alignment Network for *Stop* Signal. Traditional reinforcement learning approaches often include *stop* signal as an additional action into the action space. Nevertheless, we design an alignment network to predict a confidence score C_t for enabling the agent to have the idea of when to stop. The optimization of the alignment network can be treated as an auxiliary supervision task since the temporal IoU can explicitly provide ground-truth information for confidence score. This network is optimized by minimizing the binary cross-entropy loss between U_{t-1} and C_t:

$$\mathcal{L}_{align} = \frac{1}{M}\sum_{m=1}^{M}\sum_{t=1}^{T_{max}}[U_{t-1}\log\sigma(C_t) + (1-U_{t-1})\log(1-\sigma(C_t))]. \tag{14}$$

During testing, the agent will interact with the environment by T_{max} steps and obtain a series of C_t. Then the agent gets the maximum of C_t, which indicates that the alignment network considers U_{t-1} has a maximal temporal IoU. So $t-1$ is the termination step. The alignment network is optimized in the whole training procedure. The overall loss function in the proposed framework is summarized as:

$$\mathcal{L} = \mathcal{L}_{tree} + \lambda \mathcal{L}_{align}. \tag{15}$$

where λ is a weighting parameter to achieve a tradeoff between two types of loss.

Experiments

Datasets and Evaluation Metrics

Datasets. The models are evaluated on two widely used datasets: Charades-STA (Gao et al. 2017) and ActivityNet

			Charades-STA (Gao et al. 2017)			ActivityNet (Krishna et al. 2017)		
Paradigm	Feature	Baseline	IoU@0.7	IoU@0.5	MIoU	IoU@0.5	IoU@0.3	MIoU
SL	C3D	MCN (Anne Hendricks et al. 2017), ICCV 2017	4.44	13.66	18.77	10.17	22.07	15.99
	C3D	CTRL (Gao et al. 2017), ICCV 2017	8.89	23.63	-	14.36	29.10	21.04
	C3D	ACRN (Liu et al. 2018), SIGIR 2018	9.65	26.74	26.97	16.53	31.75	24.49
	C3D	TGN (Chen et al. 2018a), EMNLP 2018	-	-	-	28.47	45.51	-
	C3D	MAC (Ge et al. 2019), WACV 2019	12.23	29.39	29.01	-	-	-
	C3D	SAP (Chen and Jiang 2019), AAAI 2019	13.36	27.42	28.56	-	-	-
	C3D	QSPN (Xu et al. 2019), AAAI 2019	15.80	35.60	-	27.70	45.30	-
	C3D	ABLR (Yuan, Mei, and Zhu 2019), AAAI 2019	-	-	-	36.79	55.67	36.99
	I3D	MAN (Zhang et al. 2019), CVPR 2019	22.72	**46.53**	-	-	-	-
RL	C3D	SM-RL (Wang, Huang, and Wang 2019), CVPR 2019	11.17	24.36	32.22	-	-	-
	C3D	TripNet (Hahn et al. 2019), CVPRW 2019	14.50	36.61	-	32.19	48.42	-
	C3D	RWM (He et al. 2019), AAAI 2019	13.74	34.12	35.09	34.91	53.00	36.25
	C3D	TSP-PRL (Ours)	17.69	37.39	37.22	**38.76**	**56.08**	**39.21**
	Two-Stream	RWM (He et al. 2019), AAAI 2019	17.72	37.23	36.29	-	-	-
	Two-Stream	TSP-PRL (Ours)	**24.73**	45.30	**40.93**	-	-	-

Table 1: The comparison performance (in %) with state-of-the-art methods. The approaches in the first group are supervised learning (SL) based approaches and methods of the second group are reinforcement learning (RL) based approaches. "-" indicates that the corresponding values are not available.

(Krishna et al. 2017). Gao *et al.* (Gao et al. 2017) extended the original Charades dataset (Sigurdsson et al. 2016) to generate sentence-clip annotations and created the Charades-STA dataset, which comprises 12,408 sentence-clip pairs for training, and 3,720 for testing. The average duration of the videos is 29.8 seconds and the described temporally annotated clips are 8 seconds long on average. ActivityNet (Krishna et al. 2017) contains 37,421 and 17,505 video-sentence pairs for training and testing. The videos in ActivityNet are 2 minutes long on average and the described temporally annotated clips are 36 seconds long on average. ActivityNet dataset is introduced to validate the robustness of the proposed algorithm toward longer and more diverse videos.

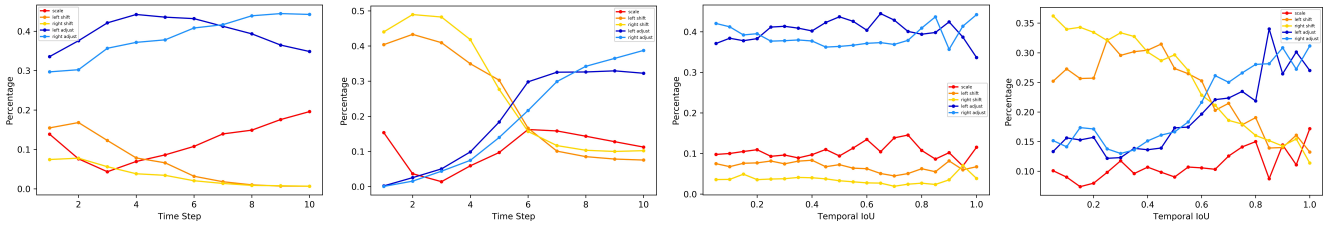
Evaluation Metrics. Following previous works (Gao et al. 2017; Yuan, Mei, and Zhu 2019), we adopt two metrics to evaluate the model for this task. "IoU@ ϵ" means the percentage of the sentence queries which have temporal IoU larger than ϵ. "MIoU" denotes the average IoU for all the sentence queries.

Implementation Details

The initial boundary is set to $L_0 = [N/4; 3N/4]$, where N denotes the clips numbers of the video. $N/4$ and $3N/4$ denote the start and end clip indices of the boundary respectively. The parameters Z is set to 16 and 80 respectively for Charades-STA and ActivityNet Datasets. We utilize two mainstream structures of action classifiers (i.e., C3D (Tran et al. 2015) and Two-Stream (Wang et al. 2016)) for video feature extraction on Charades-STA dataset. For ActivityNet, we merely employ C3D model to verify the general applicability of the proposed approach. The size of the hidden state in GRU is set to 1024. In the training stage of TSP-PRL, the batch size is set to 16 and the learning rate is 0.001 with Adam optimizer. The factor ζ is fixed to 1 in the reward settings. The hyper-parameters α, γ and λ is fixed to 0.1, 0.4 and 1 receptively. For all experiments in this paper, we use K = 200 in TSP-PRL. T_{max} is set to 20 to achieve the best trade off between accuracy and efficiency in the procedure of training and testing.

Experimental Results

Comparison with the state-of-the-art algorithms. In this subsection, we compare TSP-PRL with 12 existing state-of-the-art methods on the Charades-STA and ActivityNet datasets in Table 1. We re-implement ACRN (Liu et al. 2018), MAC (Ge et al. 2019) and RWM (He et al. 2019) and show their performance results in our experiments. The results of other approaches are taken from their paper. The well-performing methods, such as QSPN (Xu et al. 2019), ABLR (Yuan, Mei, and Zhu 2019) and MAN (Zhang et al. 2019) all delve deep into the multi-modal features representation and fusion between the verbal and visual modalities. Our approach focuses more on localization optimization, and it is complementary to the above-mentioned feature modeling methods actually. On the one hand, TSP-PRL consistently outperforms these state-of-the-art methods, w.r.t all metrics with C3D feature. For example, our method improves IoU@0.7 by 1.89% compared with the previous best (Xu et al. 2019) on the Charades-STA. For ActivityNet, the MIoU of TSP-PRL achieves the comparative enhancement over ABLR by 6.0%. MAN (Zhang et al. 2019) employs stronger I3D (Carreira and Zisserman 2017) to extract video features and obtain outstanding performance. Our method with the Two-Stream feature manages to improve IoU@0.7 from 22.72% to 24,73% on the Charades-STA. On the other hand, TSP-PRL manages to obtain more flexible boundary, avoiding exhaustive sliding window searching compared with the supervised learning-based (SL) methods. SL methods are easy to suffer from overfitting and address this task like a black-box that lack of interpretability. While TSP-PRL contributes to achieving more efficient, impressive and heuristic grounding results.



(a) "Ours w/o TSP-10" with T grows (b) "Ours-10" with T grows (c) "Ours w/o TSP-10" with IoU increases (d) "Ours-10" with IoU increases

Figure 4: The proportion curve of the selected semantic branch as time step (T) grows and IoU increases. Correspondence between line color and semantic branch: 1) red : scale branch; 2) orange: left shift branch; 3) yellow: right shift branch; 4) dark blue: left adjust branch; 5) light blue: right adjust branch. Best viewed in color.

Datasets	Charades-STA		ActivityNet	
Metrics	IoU@0.7	IoU@0.5	IoU@0.5	IoU@0.3
Ours w/o TSP-10	17.13	38.06	32.09	49.35
Ours w/o TSP-20	20.67	41.31	34.39	51.96
Ours w/o TSP-30	22.40	43.38	35.32	52.77
Ours w/o IR	20.35	40.64	35.03	52.64
Ours w/o ER	23.18	44.41	37.20	55.78
Ours w/o AN	19.03	39.78	33.89	51.03
Ours-10	22.85	44.24	37.53	55.17
Ours-20	24.73	45.30	38.76	56.08
Ours-30	**24.75**	**45.45**	**38.82**	**56.02**

Table 2: Comparison of the metrics (in %) of the proposed approach and four variants of our approach. "-j" denotes that we set the max episode lengths to j during testing.

Ablative Study

As shown in Table 2, we perform extensive ablation studies and demonstrate the effects of several essential components in our framework. The Charades-STA dataset adopts the Two-stream based feature and the ActivityNet dataset uses the C3D based feature.

Analysis of Tree-Structured Policy. To validate the significance of the tree-structured policy, we design the flat policy (denote as "Ours w/o TSP") that removes the tree-structured policy in our approach and directly maps state feature into a primitive action. As shown in Table 2, flat policy declines IoU@0.7 to 17.13%, 20.67%, and 22.40% at each level of T_{max}, with a decrease of 5.72%, 4.06%, and 2.35% when compared with our approach. Furthermore, it's performance suffers from a significant drop as T_{max} decreases, which reveals that the flat policy relies heavily on the episode lengths to obtain better results. However, our approach manages to achieve outstanding performance with fewer steps.

In order to further explore whether the tree-structured policy can better perceive environment state and decompose complex policies, we summarize the proportion of the selected high-level semantic branch at each time step and IoU interval (0.05 for each interval). The percentage curves of two models ("Ours w/o TSP-10" and "Ours-10") are depicted in Figure 4. We can observe that the flat policy tends to choose the adjust based branches all the time and is not sensitive to the time step and IoU. However, our approach manages to select the shift based branches at first few steps to reduce the semantic gap faster. When the IoU increases or time step grows, the adjust based branches gradually dominant to regulate the boundary finely. Figure 4 clearly shows that tree-structured policy contributes to efficiently improving the ability to discover complex policies which can not be learned by flat policies. To sum up, it is more intuitive and heuristic to employ the tree-structured policy, which can significantly reduce the search space and provide efficient and impressive grounding results.

Analysis of Root Reward. To delve deep into the significance of each term in the root reward, we design two variants that simply remove the intrinsic reward item (denotes as "Ours w/o IR") and extrinsic reward item (denotes as "Ours w/o ER") in the definition of the root reward. As shown in Table 2, removing the intrinsic reward term leads to an noticeable drop in performance. It indicates that the extrinsic reward item can not well reflect the quality of the root policy since this term is more relevant to the selected leaf policy. "Ours w/o ER" obtains 44.41% and 37.20% on IoU@0.5 on two datasets respectively, but it is still inferior to our approach. Taking into account the direct impact (intrinsic reward) and indirect impact (extrinsic reward) simultaneously, our approach contributes to providing accurate credit assignment and obtaining a more impressive result.

Analysis of *Stop* Signal. To demonstrate the effectiveness of the alignment network for *stop* signal, we design a variant (denote as "Ours w/o AN") that removes the alignment network and directly includes the *stop* signal as an additional action into the root policy. The baseline assigns the agent a small negative reward in proportion with the step numbers. As shown in Table 2, "Ours w/o AN" gets a less prominent performance, which may be due to the fact that it is difficult to define an appropriate reward function for the *stop* signal in this task. However, our approach manages to learn the stop information with stronger supervision information via the alignment network, and it significantly increases the performance of all metrics by a large margin.

Conclusions

We formulate a novel Tree-Structured Policy based Progressive Reinforcement Learning (TSP-PRL) approach to address the task of temporally language grounding in

untrimmed videos. The tree-structured policy is invoked at each time step to reason a series of more robust primitive actions, which can sequentially regulate the temporal boundary via an iterative refinement process. The tree-structured policy is optimized by a progressive reinforcement learning paradigm, which contributes to providing the task-oriented reward setting for correct credit assignment and optimizing the overall policy mutually and progressively. Extensive experiments show that our approach achieves competitive performance over state-of-the-art methods on the widely used Charades-STA and ActivityNet datasets.

References

[Anne Hendricks et al. 2017] Anne Hendricks, L.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision*, 5803–5812.

[Carreira and Zisserman 2017] Carreira, J., and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.

[Chaplot et al. 2018] Chaplot, D. S.; Sathyendra, K. M.; Pasumarthi, R. K.; Rajagopal, D.; and Salakhutdinov, R. 2018. Gated-attention architectures for task-oriented language grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[Chen and Jiang 2019] Chen, S., and Jiang, Y.-G. 2019. Semantic proposal for activity localization in videos via sentence query. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[Chen et al. 2018a] Chen, J.; Chen, X.; Ma, L.; Jie, Z.; and Chua, T.-S. 2018a. Temporally grounding natural sentence in video. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 162–171.

[Chen et al. 2018b] Chen, T.; Wang, Z.; Li, G.; and Lin, L. 2018b. Recurrent attentional reinforcement learning for multi-label image recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Cho et al. 2014] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[Gao et al. 2017] Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision*, 5267–5275.

[Ge et al. 2019] Ge, R.; Gao, J.; Chen, K.; and Nevatia, R. 2019. Mac: Mining activity concepts for language-based temporal localization. In *IEEE Winter Conference on Applications of Computer Vision*, 245–253. IEEE.

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

[He et al. 2019] He, D.; Zhao, X.; Huang, J.; Li, F.; Liu, X.; and Wen, S. 2019. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[Kiros et al. 2015] Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, 3294–3302.

[Krishna et al. 2017] Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 706–715.

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

[Mancas et al. 2016] Mancas, M.; Ferrera, V. P.; Riche, N.; and Taylor, J. G. 2016. *From Human Attention to Computational Attention*, volume 2. Springer.

[Shi et al. 2019] Shi, Y.; Guanbin, L.; Cao, Q.; Wang, K.; and Lin, L. 2019. Face hallucination by attentive sequence optimization with reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*.

[Sigurdsson et al. 2016] 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 *European Conference on Computer Vision*, 510–526. Springer.

[Sutton and Barto 2018] Sutton, R. S., and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

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

[Wang et al. 2016] Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, 20–36. Springer.

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

[Williams 1992] Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*. Springer. 5–32.

[Wu et al. 2019a] Wu, J.; Chen, T.; Wu, H.; Yang, Z.; Wang, Q.; and Lin, L. 2019a. Concrete image captioning by integrating content sensitive and global discriminative objective. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 1306–1311. IEEE.

[Wu et al. 2019b] Wu, W.; He, D.; Tan, X.; Chen, S.; and Wen, S. 2019b. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 6222–6231.

[Xia et al. 2017] Xia, Y.; Tian, F.; Wu, L.; Lin, J.; Qin, T.; Yu, N.; and Liu, T.-Y. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc. 1784–1794.

[Xu et al. 2019] Xu, H.; He, K.; Sigal, L.; Sclaroff, S.; and Saenko, K. 2019. Multilevel language and vision integration for text-to-clip retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 2, 7.

[Yeung et al. 2016] Yeung, S.; Russakovsky, O.; Mori, G.; and Fei-Fei, L. 2016. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2678–2687.

[Yuan, Mei, and Zhu 2019] Yuan, Y.; Mei, T.; and Zhu, W. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[Zhang et al. 2019] Zhang, D.; Dai, X.; Wang, X.; Wang, Y.-F.; and Davis, L. S. 2019. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1247–1257.